Using the Bay Command Console (BCC™)

BayRS Version 12.20
Site Manager Software Version 6.20

BCC 4.00

Part No. 117383-B Rev. 00
June 1998

Bay Networks

Bay Networks

4401 Great America Parkway
Santa Clara, CA 95054

8 Federal Street
Billerica, MA 01821

Copyright © 1998 Bay Networks, Inc.

All rights reserved. Printed in the USA. June 1998.

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Bay Networks, Inc.

The software described in this document is furnished under a license agreement and may only be used in accordance with the terms of that license. A summary of the Software License is included in this document.

Trademarks

AN, BCN, BLN, BN, FRE, and Bay Networks are registered trademarks and Advanced Remote Node, ANH, ARN, ASN, BayRS, BCC, System 5000, and the Bay Networks logo are trademarks of Bay Networks, Inc. All other trademarks and registered trademarks are the property of their respective owners.

Restricted Rights Legend

Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Notwithstanding any other license agreement that may pertain to, or accompany the delivery of, this computer software, the rights of the United States Government regarding its use, reproduction, and disclosure are as set forth in the Commercial Computer Software-Restricted Rights clause at FAR 52.227-19.

Statement of Conditions

In the interest of improving internal design, operational function, and/or reliability, Bay Networks, Inc. reserves the right to make changes to the products described in this document without notice.

Bay Networks, Inc. does not assume any liability that may occur due to the use or application of the product(s) or circuit layout(s) described herein.

Portions of the code in this software product may be Copyright © 1988, Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms of such portions are permitted, provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that such portions of the software were developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from such portions of the software without specific prior written permission.

SUCH PORTIONS OF THE SOFTWARE ARE PROVIDED “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

In addition, the program and information contained herein are licensed only pursuant to a license agreement that contains restrictions on use and disclosure (that may incorporate by reference certain limitations and notices imposed by third parties).

USA Requirements Only

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy. If it is not installed and used in accordance with the instruction manual, it may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to take whatever measures may be necessary to correct the interference at their own expense.

European Requirements Only

EN 55 022 Statement

This is to certify that the Bay Networks <product or system name> is shielded against the generation of radio interference in accordance with the application of Council Directive 89/336/EEC, Article 4a. Conformity is declared by the application of EN 55 022 Class A (CISPR 22).

Warning: This is a Class A product. In a domestic environment, this product may cause radio interference, in which case, the user may be required to take appropriate measures.

Japan/Nippon Requirements Only

Voluntary Control Council for Interference (VCCI) Statement

この装置は、情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

Canada Requirements Only

Canadian Department of Communications Radio Interference Regulations

This digital apparatus (<product or system name>) does not exceed the Class A limits for radio-noise emissions from digital apparatus as set out in the Radio Interference Regulations of the Canadian Department of Communications.

Règlement sur le brouillage radioélectrique du ministère des Communications

Cet appareil numérique (<product or system name>) respecte les limites de bruits radioélectriques visant les appareils numériques de classe A prescrites dans le Règlement sur le brouillage radioélectrique du ministère des Communications du Canada.

Bay Networks, Inc. Software License Agreement

NOTICE: Please carefully read this license agreement before copying or using the accompanying software or installing the hardware unit with pre-enabled software (each of which is referred to as “Software” in this Agreement). BY COPYING OR USING THE SOFTWARE, YOU ACCEPT ALL OF THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT. THE TERMS EXPRESSED IN THIS AGREEMENT ARE THE ONLY TERMS UNDER WHICH BAY NETWORKS WILL PERMIT YOU TO USE THE SOFTWARE. If you do not accept these

terms and conditions, return the product, unused and in the original shipping container, within 30 days of purchase to obtain a credit for the full purchase price.

1. License Grant. Bay Networks, Inc. (“Bay Networks”) grants the end user of the Software (“Licensee”) a personal, nonexclusive, nontransferable license: a) to use the Software either on a single computer or, if applicable, on a single authorized device identified by host ID, for which it was originally acquired; b) to copy the Software solely for backup purposes in support of authorized use of the Software; and c) to use and copy the associated user manual solely in support of authorized use of the Software by Licensee. This license applies to the Software only and does not extend to Bay Networks Agent software or other Bay Networks software products. Bay Networks Agent software or other Bay Networks software products are licensed for use under the terms of the applicable Bay Networks, Inc. Software License Agreement that accompanies such software and upon payment by the end user of the applicable license fees for such software.

2. Restrictions on use; reservation of rights. The Software and user manuals are protected under copyright laws. Bay Networks and/or its licensors retain all title and ownership in both the Software and user manuals, including any revisions made by Bay Networks or its licensors. The copyright notice must be reproduced and included with any copy of any portion of the Software or user manuals. Licensee may not modify, translate, decompile, disassemble, use for any competitive analysis, reverse engineer, distribute, or create derivative works from the Software or user manuals or any copy, in whole or in part. Except as expressly provided in this Agreement, Licensee may not copy or transfer the Software or user manuals, in whole or in part. The Software and user manuals embody Bay Networks’ and its licensors’ confidential and proprietary intellectual property. Licensee shall not sublicense, assign, or otherwise disclose to any third party the Software, or any information about the operation, design, performance, or implementation of the Software and user manuals that is confidential to Bay Networks and its licensors; however, Licensee may grant permission to its consultants, subcontractors, and agents to use the Software at Licensee’s facility, provided they have agreed to use the Software only in accordance with the terms of this license.

3. Limited warranty. Bay Networks warrants each item of Software, as delivered by Bay Networks and properly installed and operated on Bay Networks hardware or other equipment it is originally licensed for, to function substantially as described in its accompanying user manual during its warranty period, which begins on the date Software is first shipped to Licensee. If any item of Software fails to so function during its warranty period, as the sole remedy Bay Networks will at its discretion provide a suitable fix, patch, or workaround for the problem that may be included in a future Software release. Bay Networks further warrants to Licensee that the media on which the Software is provided will be free from defects in materials and workmanship under normal use for a period of 90 days from the date Software is first shipped to Licensee. Bay Networks will replace defective media at no charge if it is returned to Bay Networks during the warranty period along with proof of the date of shipment. This warranty does not apply if the media has been damaged as a result of accident, misuse, or abuse. The Licensee assumes all responsibility for selection of the Software to achieve Licensee’s intended results and for the installation, use, and results obtained from the Software. Bay Networks does not warrant a) that the functions contained in the software will meet the Licensee’s requirements, b) that the Software will operate in the hardware or software combinations that the Licensee may select, c) that the operation of the Software will be uninterrupted or error free, or d) that all defects in the operation of the Software will be corrected. Bay Networks is not obligated to remedy any Software defect that cannot be reproduced with the latest Software release. These warranties do not apply to the Software if it has been (i) altered, except by Bay Networks or in accordance with its instructions; (ii) used in conjunction with another vendor’s product, resulting in the defect; or (iii) damaged by improper environment, abuse, misuse, accident, or negligence. THE FOREGOING WARRANTIES AND LIMITATIONS ARE EXCLUSIVE REMEDIES AND ARE IN LIEU OF ALL OTHER WARRANTIES EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Licensee is responsible for the security of its own data and information and for maintaining adequate procedures apart from the Software to reconstruct lost or altered files, data, or programs.

4. Limitation of liability. IN NO EVENT WILL BAY NETWORKS OR ITS LICENSORS BE LIABLE FOR ANY COST OF SUBSTITUTE PROCUREMENT; SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES; OR ANY DAMAGES RESULTING FROM INACCURATE OR LOST DATA OR LOSS OF USE OR PROFITS ARISING OUT OF OR IN CONNECTION WITH THE PERFORMANCE OF THE SOFTWARE, EVEN IF BAY NETWORKS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT

SHALL THE LIABILITY OF BAY NETWORKS RELATING TO THE SOFTWARE OR THIS AGREEMENT EXCEED THE PRICE PAID TO BAY NETWORKS FOR THE SOFTWARE LICENSE.

5. Government Licensees. This provision applies to all Software and documentation acquired directly or indirectly by or on behalf of the United States Government. The Software and documentation are commercial products, licensed on the open market at market prices, and were developed entirely at private expense and without the use of any U.S. Government funds. The license to the U.S. Government is granted only with restricted rights, and use, duplication, or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c)(1) of the Commercial Computer Software—Restricted Rights clause of FAR 52.227-19 and the limitations set out in this license for civilian agencies, and subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause of DFARS 252.227-7013, for agencies of the Department of Defense or their successors, whichever is applicable.

6. Use of Software in the European Community. This provision applies to all Software acquired for use within the European Community. If Licensee uses the Software within a country in the European Community, the Software Directive enacted by the Council of European Communities Directive dated 14 May, 1991, will apply to the examination of the Software to facilitate interoperability. Licensee agrees to notify Bay Networks of any such intended examination of the Software and may procure support and assistance from Bay Networks.

7. Term and termination. This license is effective until terminated; however, all of the restrictions with respect to Bay Networks' copyright in the Software and user manuals will cease being effective at the date of expiration of the Bay Networks copyright; those restrictions relating to use and disclosure of Bay Networks' confidential information shall continue in effect. Licensee may terminate this license at any time. The license will automatically terminate if Licensee fails to comply with any of the terms and conditions of the license. Upon termination for any reason, Licensee will immediately destroy or return to Bay Networks the Software, user manuals, and all copies. Bay Networks is not liable to Licensee for damages in any form solely by reason of the termination of this license.

8. Export and Re-export. Licensee agrees not to export, directly or indirectly, the Software or related technical data or information without first obtaining any required export licenses or other governmental approvals. Without limiting the foregoing, Licensee, on behalf of itself and its subsidiaries and affiliates, agrees that it will not, without first obtaining all export licenses and approvals required by the U.S. Government: (i) export, re-export, transfer, or divert any such Software or technical data, or any direct product thereof, to any country to which such exports or re-exports are restricted or embargoed under United States export control laws and regulations, or to any national or resident of such restricted or embargoed countries; or (ii) provide the Software or related technical data or information to any military end user or for any military end use, including the design, development, or production of any chemical, nuclear, or biological weapons.

9. General. If any provision of this Agreement is held to be invalid or unenforceable by a court of competent jurisdiction, the remainder of the provisions of this Agreement shall remain in full force and effect. This Agreement will be governed by the laws of the state of California.

Should you have any questions concerning this Agreement, contact Bay Networks, Inc., 4401 Great America Parkway, P.O. Box 58185, Santa Clara, California 95054-8185.

LICENSEE ACKNOWLEDGES THAT LICENSEE HAS READ THIS AGREEMENT, UNDERSTANDS IT, AND AGREES TO BE BOUND BY ITS TERMS AND CONDITIONS. LICENSEE FURTHER AGREES THAT THIS AGREEMENT IS THE ENTIRE AND EXCLUSIVE AGREEMENT BETWEEN BAY NETWORKS AND LICENSEE, WHICH SUPERSEDES ALL PRIOR ORAL AND WRITTEN AGREEMENTS AND COMMUNICATIONS BETWEEN THE PARTIES PERTAINING TO THE SUBJECT MATTER OF THIS AGREEMENT. NO DIFFERENT OR ADDITIONAL TERMS WILL BE ENFORCEABLE AGAINST BAY NETWORKS UNLESS BAY NETWORKS GIVES ITS EXPRESS WRITTEN CONSENT, INCLUDING AN EXPRESS WAIVER OF THE TERMS OF THIS AGREEMENT.

Contents

About This Guide

Conventions viii
Acronyms ix
Ordering Bay Networks Publications x
Bay Networks Customer Service x
How to Get Help xi

Chapter 1
Overview of the BCC Interface

Platform Requirements 1-2
Number of BCC Sessions 1-2
Configurable Objects 1-3
Terminology and Concepts 1-3
Using Abbreviations and Acronyms 1-9

Chapter 2
Learning to Use the BCC Interface

Entering and Exiting the BCC Interface 2-1
Using the BCC Configuration Hierarchy 2-3
Locating Configured Objects 2-5
Displaying Context 2-6
Context-Sensitive Prompts 2-6
Navigating in Configuration Mode 2-7
Navigating with the back Command 2-7
Navigating with Configuration Commands 2-8
Moving Back One or More Levels 2-8
Moving Back to Root Level 2-8
Moving Forward from Root (box) Level 2-9
Specifying a Full Path 2-9
Specifying a Shortened Path 2-10

Displaying Online Help 2-12
Using the BCC Help File 2-22
Displaying Help File Version Data 2-23
Displaying Configuration Data 2-23
Displaying the Total Device Configuration 2-23
Displaying the Configuration of One Object 2-25
Displaying Binary Configuration Files as BCC Syntax 2-26
Saving show config Output to an ASCII file on the Router 2-26
Saving the Active Configuration as a Bootable Binary File 2-26
Saving Commands on a Workstation 2-27

Chapter 3
Entering Commands

Command Input Features 3-2
Using Command Abbreviations 3-2
Recalling Commands 3-2
Entering Commands from a File 3-3
Entering Multiple Commands per Line 3-4
Continuing a Command Line (Underscore Prompt) 3-4
Entering Comments 3-6
System Commands 3-6
Configuration Commands 3-7
Using Basic (Full) Syntax 3-7
Using Default Syntax 3-8
Using Abbreviated Syntax 3-9
Specifying Parameter Values 3-10
Required, Derived, and Other Parameters 3-10
Specifying Multiple Parameter-Value Pairs 3-11
Command Operators 3-11
Editing Command Lines 3-13

Chapter 4
Configuring a Network Device

Creating and Modifying a Device Configuration 4-1
Entering Configuration Commands from a File 4-15
Disabling a Configured Object 4-16
Enabling a Configured Object 4-17
Deleting a Configured Object 4-18
Configuration Command Responses 4-19

Appendix A
System Commands

Appendix B
TCL Support

Appendix C
System-Oriented Show Commands

show console C-2
show hardware C-5
show process C-7
show system C-10

Figures

Figure 1-1. The Technician Interface and the BCC Interface 1-1
Figure 1-2. Configuring IP and RIP on an Ethernet Interface 1-5
Figure 2-1. Sample BCC Configuration (BN Router) 2-4
Figure 2-2. Object Location or Context 2-5
Figure 2-3. Moving Away from Root (box) Level 2-9
Figure 2-4. Specifying a Full Path 2-10
Figure 2-5. Navigating with the BCC Backward (Recursive) Search Feature 2-11
Figure 2-6. Objects to Configure at the Next (Branch) Level 2-21
Figure 4-1. Sample BCC Configuration (BCN Router) 4-3
Figure 4-2. Typical BCC Configuration Cycle 4-4

Tables

Table 3-1. BCC Command Operators 3-12
Table 3-2. Keystrokes for Editing BCC Command Lines 3-13
Table A-1. System Commands A-1

About This Guide

If you are responsible for configuring and managing Bay Networks® AN®, ANH, ARN, ASN, BN® (BCN® and BLN®), and System 5000 routers, read this guide to learn how to use the Bay Command Console (BCC™).

If you want to	Go to
Learn about BCC concepts and terminology	Chapter 1
Learn how to perform basic BCC operations	Chapter 2
Learn how to enter BCC commands	Chapter 3
Learn how to use the BCC to configure a router running BayRS software	Chapter 4

To use the BCC, you should have some experience supporting a multivendor internetworking system. You should be able to perform network device configuration, maintenance, and troubleshooting.

For additional sources of information about configuring Bay Networks routers, see the task map for this release.

Note: Because the BCC makes real-time changes to a device configuration, Bay Networks recommends that you first learn about BCC behavior on a device not connected to your production network. After you become comfortable with using the BCC, run it on a device in your production network.

Conventions

angle brackets (< >)	Indicate a variable in a command line. The word or words inside the angle brackets describe the type of variable (for example, <*host-address*>). Do not type the angle brackets when entering an actual value for a variable. Example: if command syntax is **ping** <*ip-address*>, enter **ping 192.32.10.12**
`bold text`	Indicates text (usually commands) that you enter at the BCC command line prompt. Example: Enter **`reset`**.

braces ({ })	Enclose a list of
	
	• Command keywords or configuration parameter values, one of which you can enter
	
	• BCC instance IDs of configured objects
	
	Example:
	`{arp/1.2.3.4/1 rip/1.2.3.4}`
	
	• A literal string, such as the literal value of the console prompt
	
	Example:
	`prompt {[%slot%:1]$}`
	
	(Yields a prompt such as "`[9:1]$`".)

brackets ([ ])	Indicate command keywords, arguments, or filters not required (taken as optional command input) by the BCC.
ellipsis points (. . .)	Repeat the last element of the command as needed. Example: **ethernet/2/1** [<*parameter*> <*value*>] **. . .** Translating this syntax to a specific command: **ethernet/2/1 bofl-retries 2 hardware-filter enabled**

italic text	Indicates variable values in command syntax descriptions, new terms, file and directory names, and book titles.
quotation marks (“ ”)	Indicate a literal string in a command line, the title of a chapter, or a section within a book.
`screen text`	Indicates BCC or Technician Interface output to a console or Telnet screen, for example, prompts, system messages, statistical data, and configuration data.
vertical line (|)	Separates choices for required or optional command keywords and arguments. You must enter only one of the choices available. Do not type the vertical line when entering a command. Example: If the command syntax is **show ip routes** | **rip**, you enter either **show ip routes** or **show ip rip**, but not both.

Acronyms

ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
BN	Backbone Node
DCM	Data Collection Module
IP	Internet Protocol
IPX	Internet Packet Exchange
LAN	local area network
MAC	media access control
OSPF	Open Shortest Path First
RIP	Routing Information Protocol
SNMP	Simple Network Management Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
WAN	wide area network

Ordering Bay Networks Publications

To purchase additional copies of this document or other Bay Networks publications, order by part number from Bay Networks Press™ at the following numbers:

- Phone--U.S./Canada: 888-422-9773
- Phone--International: 510-490-4752
- FAX--U.S./Canada and International: 510-498-2609

The Bay Networks Press catalog is available on the World Wide Web at *support.baynetworks.com/Library/GenMisc*. Bay Networks publications are available on the World Wide Web at *support.baynetworks.com/Library/tpubs*.

Bay Networks Customer Service

You can purchase a support contract from your Bay Networks distributor or authorized reseller, or directly from Bay Networks Services. For information about, or to purchase a Bay Networks service contract, either call your local Bay Networks field sales office or one of the following numbers:

Region	Telephone number	Fax number
United States and Canada	800-2LANWAN; then enter Express Routing Code (ERC) 290, when prompted, to purchase or renew a service contract 978-916-8880 (direct)	978-916-3514
Europe	33-4-92-96-69-66	33-4-92-96-69-96
Asia/Pacific	61-2-9927-8888	61-2-9927-8899
Latin America	561-988-7661	561-988-7550

Information about customer service is also available on the World Wide Web at *support.baynetworks.com*.

How to Get Help

If you purchased a service contract for your Bay Networks product from a distributor or authorized reseller, contact the technical support staff for that distributor or reseller for assistance.

If you purchased a Bay Networks service program, call one of the following Bay Networks Technical Solutions Centers:

Technical Solutions Center	Telephone number	Fax number
Billerica, MA	800-2LANWAN	978-916-3514
Santa Clara, CA	800-2LANWAN	408-495-1188
Valbonne, France	33-4-92-96-69-68	33-4-92-96-69-98
Sydney, Australia	61-2-9927-8800	61-2-9927-8811
Tokyo, Japan	81-3-5402-0180	81-3-5402-0173

Chapter 1
Overview of the BCC Interface

The BCC is a command-line interface for configuring Bay Networks devices. After logging on to a device, you access the BCC by entering the **bcc** command at the Technician Interface prompt (Figure 1-1).

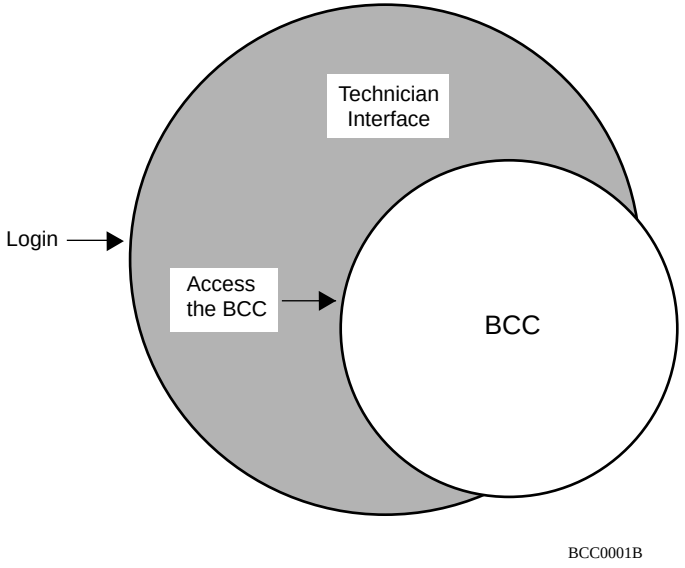

Figure 1-1. The Technician Interface and the BCC Interface

From the bcc> prompt, you can

- Run any system command not classified as "Technician Interface only." (See Appendix A, "System Commands.")
- Use configuration mode to perform tasks such as creating or deleting IP interfaces on the router. Enter BCC configuration mode by entering the **config** command at the bcc> prompt.

For tasks related to device management (managing files on the router, viewing statistics or the event log, and so on), you can enter Technician Interface commands directly at the BCC command-line prompt. (For more information about Technician Interface commands, refer to *Using Technician Interface Software.*)

Platform Requirements

The BCC software requires 16 MB of dynamic RAM (DRAM) to run on any (AN/ANH, ARN, ASN, BCN, BLN, and System 5000) router platform. In addition, a minimum of 2.0 MB of DRAM must be available when you start the first BCC session on a slot.

Number of BCC Sessions

You can open one BCC session per slot in read-write (configuration) mode. Other users can open additional BCC sessions in read-only (nonconfiguration) mode on the same slot, depending on available memory. (When memory resources are insufficient to support additional BCC sessions on a login slot, the system displays the following message:

"BCC requires 2 MB of free memory. Current available memory is <*size*>."

Separate read-write sessions on different slots have independent views of the current configuration. (The BCC does not dynamically update the configuration view in one session, based on configuration changes made in another session.)

Configurable Objects

Refer to the *Release Notes* for the most up-to-date information about services you can configure using the BCC on any (AN/ANH, ARN, ASN, BCN, BLN, or System 5000) platform. You can obtain a complete hierarchical listing of all objects configurable on a device by entering the **help tree -all** command at any BCC prompt.

Terminology and Concepts

The following list of terms and concepts defines key functional and user-oriented aspects of the BCC interface:

Object -- A data structure representing a configurable physical or logical entity such as a physical interface or a protocol on an interface. Every configurable object belongs to a specific *class* that defines its characteristics.

Class -- A template for a configurable object (such as Ethernet or the protocol IP). When you add a new object to the configuration of a device, the BCC creates a copy (an *instance*) of the appropriate template. Each instance is an object uniquely identifiable within the total device configuration.

Parameters -- Attributes or properties of a configurable object. For example, some parameters of Ethernet on a BN platform are

- slot and connector (describing the location of the Ethernet interface)
- bofl-retries (describing one behavioral aspect of the Ethernet interface)

Required parameters -- A minimum set of parameters for which the BCC requires you to supply values in order to configure the object. For example:

- The parameters you must specify to configure any physical port (ethernet, serial, etc.) on a BN platform are **slot** and **connector**
- The required parameters of a physical port on an ASN platform are **slot**, **module**, and **connector**.

Derived parameters -- Parameters for which the BCC supplies a value. For example, a derived parameter of OSPF (global) is **router-id**. In this case, the BCC derives a value for router-id from the address of the first IP interface configured on the device.

Optional parameters -- The set of parameters for which you can optionally specify customized values, replacing any default values set by the system. For example, an optional parameter of an Ethernet interface is **bofl-retries**. This parameter normally has a system default value of **5** (5 retries), but you can change this to another numeric value.

BCC instance identifier -- Uniquely identifies a single instance of an object configured on a Bay Networks device. The BCC instance ID consists typically of the name of an object, combined with the values you specify for its required parameters. For example:

- The BCC instance ID for an Ethernet interface on a BN platform consists of **ethernet/**<*slot*>**/**<*connector*>, as in **ethernet/2/1**.
- The BCC instance ID for an Ethernet interface on an ASN platform consists of **ethernet/**<*slot*>**/**<*module*>**/**<*connector*>, as in **ethernet/1/2/1**.

For some objects, the BCC automatically appends an internally generated number to ensure uniqueness of the BCC instance ID. For example, the BCC creates an instance ID for ARP on IP based on the object name (arp), plus the address of the underlying IP interface (1.2.3.4), plus an internally generated integer, resulting in an ID such as **arp/1.2.3.4/1**.

In other cases, a configurable object may also have required parameters that do not become part of its BCC instance ID. For example, the global OSPF object has a required *router-id* parameter that does not become part of the instance ID. (Many such required parameters end in "-id," such as *global-id* and *router-id*.)

Each object has its own requirements for unique instance identification within the total device configuration.

Configuration hierarchy -- Classes (templates for creating objects) exist within a tree hierarchy. Just as a file system has a root directory, subdirectores, and files, the BCC configuration system has a root-level object ("box" or "stack") and branch objects such as interfaces and protocols fan out from root level in a tree hierarchy. (The root level object for AN/ANH, ARN, and BN platforms is "box"; the root-level object for ASN and System 5000 platforms is "stack".)

The BCC configuration command hierarchy varies according to the type of network device (for example, a particular type of router, hub, or switch). However, BCC commands, such as **?**, **help tree**, and **back**, enable you to efficiently discover and navigate the configuration hierarchy. (You can view the configuration hierarchy by entering the **help tree -all** command at any BCC prompt.)

You create a customized configuration by using BCC commands to create new objects, and to modify or delete objects in an existing configuration. You begin at root level in BCC configuration mode and navigate to (or create new) objects in the device configuration tree.

For example, with a BLN router, you can use BCC commands to add a new physical interface (such as Ethernet) on *box*, add IP to the Ethernet interface, and then add RIP to IP on that interface. Figure 1-2 shows the sequence of commands necessary to build this configuration.

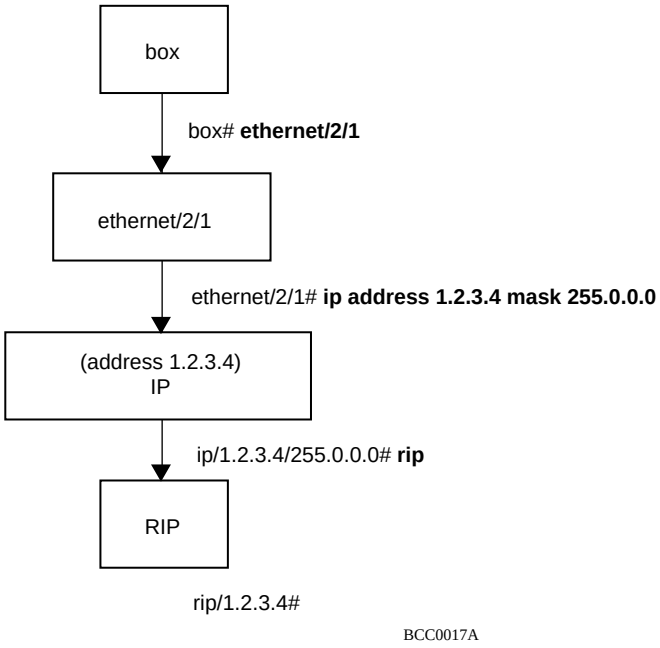

Figure 1-2. Configuring IP and RIP on an Ethernet Interface

Context -- Your working *location* or *level* within the BCC configuration tree. Just as a UNIX file system has a current working directory within which you can add, modify, or delete files, the BCC configuration tree has a current working *context*, within which you can add, modify, disable, reenable, or delete objects.

The BCC indicates *context* by displaying a context-sensitive prompt in configuration mode. The prompt identifies the configured object at your current working location or level within the configuration hierarchy. For example, after logging into a BLN router as Manager, then configuring or navigating to the Ethernet interface on slot 2 connector 1, the BCC displays the following prompt:

```
ethernet/2/1#
```

Global/Box-wide objects -- Objects that provide services uniformly to all slots of a network device (box-wide); for example, global IP, BGP, TCP, SNMP, FTP, NTP, TFTP, and Telnet. Some protocols, such as IP, RIP, and OSPF, have global and interface-level objects.

Board -- The BCC uses the name *board* to identify any logic or circuit board in a Bay Networks device. Each board occupies a *slot* in a network device. On some Bay Networks products, one board may contain another board such as an RMON probe or a data collection module (DCM). All board objects have an associated *type* parameter. For example, “qenf” (Quad Ethernet with Filters) identifies the type of a board (or module) in a specific slot on a BN router.

Note: Refer to the *Release Notes* for module descriptions based on the literal value of the *type* parameter for any board object.

Slot -- The physical position of a module installed in a network device. Bay Networks has single-slot, single-slot stackable, and multislot devices.

Multislot devices such as the BLN or BCN router accommodate one system (SRML) or link (interface) module per slot.

Single-slot devices such as the AN, ASN and ARN router accommodate one base module (in slot 1), 1 or 2 adapter modules, and 1 expansion module.

Platform	Interface/Port Numbering
AN/ANH	***<interface> <slot>/<connector>*** • *<interface>* = Interface type: ethernet, token-ring, serial (etc.) • *<slot>* number always = 1 (AN/ANH is a 1-slot device) • *<connector>* numbering starts with connector 1.) ***Example:*** **ethernet slot 1 connector 2** This is an Ethernet interface configured on AN/ANH connector 2, which exists on an Ethernet adapter module. (Connectors 1 and 2 are on the base.)
ASN	***<interface> <slot>/<module>/<connector>*** • *<interface>* = Interface type: ethernet, token-ring, serial, (etc.) • *<slot_number>* = 1, 2, 3, or 4, depending on the manual setting of the Module ID switch on each ASN box. The switch indicates the position of specific ASN chassis in a stackable ASN configuration. • *<module>* numbering corresponds to Net Module numbering (1 to 4) on each ASN box. • *<connector>* numbering starts at 1 on each Net Module (per media type) ***Example:*** **ethernet slot 2 module 3 connector 2** This is an Ethernet interface on connector 2 of Net Module 3, in ASN box 2 of a stacked ASN configuration.
ARN	***<interface> <slot>/<connector>*** • *<interface>* = Interface type: ethernet, token-ring, serial (etc.) • *<slot>* number always = 1 • *<connector>* numbering depends on the port type (LAN or WAN). LAN connector numbering starts at 1 on the base module, which contains only LAN ports. LAN connector numbering continues in ascending order, starting with the first LAN port on an ARN expansion module. (The ARN expansion module plugs into the ARN base module.) WAN connector numbering starts with connector 1 on WAN Adapter module 1, continues with connector 2 on WAN Adapter module 2, and ascends sequentially with WAN connectors 3 through *n* on the ARN expansion module. ***Example:*** **ethernet/1/2** This interface is configured on LAN connector 2, which exists physically on an Ethernet expansion module. (Ethernet connector 1 is on the base module.) ***Example:*** **serial/1/3** This is a serial (WAN) interface configured on WAN connector 3, which exists physically on the ARN expansion module.

Platform	Interface/Port Numbering
BN	***<interface> <slot>/<connector>*** • *<interface>* = Interface type: ethernet, token-ring, serial (etc.) • *<slot>* number = 2 to 5 (BLN) or 1 to 14 (BCN). • *<connector>* numbering starts with connector 1 on each slot (per media type). ***Example:*** **ethernet slot 8 connector 3** This is an Ethernet interface configured on connector 3 of slot 8 of a BCN router.
System 5000	***<interface> <slot>/<module>/<connector>*** • *<interface>* = Interface type: ethernet, token-ring, serial, atm, (etc.) • *<slot_number>* = 2 to 13 in a System 5000 chassis • *<module>* numbering corresponds to Net Modules 1 or 2, which plug into a System 5000 base module. Any System 5000 base module is always module 3. • *<connector>* numbering starts at 1 on each Net Module. The base module always contains a connector 1, which plugs into a System 5000 backplane (for example, Ethernet, Token Ring, or ATM backplane. ***Example:*** **ethernet slot 2 module 2 connector 1** **This is an Ethernet interface on connector 1 of the Ethernet Net Module (module 2) in slot 2.** **atm slot 2 module 3 connector 1** This is an ATM interface on connector 1 of slot 2, a Model 5782 Virtual Network Router (VNR) base module (module 3). Note that the VNR does not accommodate any Net Modules.

Module -- A board in a Bay Networks router. The board provides one of the following functions: base module, net module, adapter module, or expansion module.

Connector -- The physical and electrical or optical means to interconnect a network device directly or indirectly to a physical-layer transmission medium.

Line -- A physical (and in some cases, logical) circuit identified typically by means of a slot, connector, interface type (ethernet, sync, fddi, and so on), and, where applicable, a channel number (such as with TI/E1 interface types).

Port -- An interface object defined by its type (for example, an Ethernet port) and location (slot and connector) within a network device. On a network device or a user end station, a port is also a logical point of termination for data sent or received by a specific protocol or application. For example, a UNIX workstation receives syslog messages from a remote device at UDP logical port number 162.

Interface, physical -- Circuitry and digital logic associated with the interconnection between a physical network medium (such as Ethernet) and a higher-layer protocol entity (such as IP).

Interface, logical -- An addressable entity for originating and terminating connections across an IP network.

Protocol -- A configurable object that typically provides data link, network, transport, session, application, or management layer services on a network device. A protocol on a router may provide services box-wide (across all slots, such as global IP) or on a per-interface basis (such as interface IP) on a router platform.

System commands -- Enable you to perform system administration tasks from any command-line prompt. For a complete list of system commands available at any level of the BCC interface, enter **?** at any BCC command-line prompt.

Using Abbreviations and Acronyms

Words that represent objects, parameters, and certain parameter values for command input or output are

- Industry-accepted words or standard abbreviations and acronyms
- Standard Bay Networks abbreviations and acronyms

For command input, the BCC interface allows you to shorten existing object and parameter names, for example, **e** or **eth** for **ethernet**. This is the “minimum to distinguish” feature of the BCC interface.

Example

Three objects that begin with “t” (**tftp**, **telnet**, and **token-ring**) exist at the root (box) level of the AN/BN router configuration tree. So that the BCC knows which of these objects you want to configure, you must minimally enter at least two letters (for example, **te** for **telnet**) at the `box#` prompt.

Chapter 2
Learning to Use the BCC Interface

This chapter provides primary instructions on how to use the BCC interface to complete any of the following tasks:

Topic	Page
Entering and Exiting the BCC Interface	page 2-1
Using the BCC Configuration Hierarchy	page 2-3
Navigating in Configuration Mode	page 2-7
Displaying Online Help	page 2-12
Using the BCC Help File	page 2-22
Displaying Configuration Data	page 2-23

Entering and Exiting the BCC Interface

To access the BCC interface on a Bay Networks router, first open a Technician Interface session with the target device from

- An ASCII terminal (for example, a VT-100 device) locally attached to the router console port
- A workstation or PC running terminal emulation software and locally attached to the router console port
- A remote workstation or PC running Telnet

To enter and exit the BCC:

1. **Access the Technician Interface on the router by entering the Manager or User command at the `Login` prompt that appears on your Telnet or console display.**

   After you start the BCC interface, the Manager login allows you to enter any system command. A Manager login also allows, by default, read-write access to the device configuration. The User login allows you to enter only user-level system commands, and allows read-only access to the device configuration. With read-only access, you can display or navigate the existing configuration, but you cannot modify the configuration.

2. **Start the BCC interface by entering bcc at the Technician Interface prompt.**

```
Router1> bcc

Welcome to Bay Command Console!

     * To enter configuration mode, type config

     * To list all system commands, type ?

     * To exit the BCC, type exit
bcc>
```

3. **Start BCC configuration mode by entering config at the `bcc>` prompt.**

```
bcc>  config
box#
```

   You enter configuration mode at the root (box) level of the BCC configuration tree. The prompt ends with a pound (#) symbol if you have read-write privileges (Manager only), or with a greater than symbol (>) if you have read-only privileges (Manager or User).

   If you enter BCC configuration mode as Manager and want to change your privilege level for the current session from read-write to read-only, enter **config -read-only**. To change Manager privileges back to read-write, enter **config -read-write**. You cannot change your privilege level from read-only to read-write if you logged in as User.

Caution: When you enter BCC configuration commands with read-write (Manager) access privileges, you immediately modify the device configuration. This is equivalent to dynamic mode configuration with the Bay Networks Site Manager graphical user interface.

4. **When you finish using BCC configuration mode, enter the exit command at any prompt.**

```
box#  exit
bcc>
```

5. **When you finish using the BCC, enter the exit command at the** `bcc>` **prompt.**

```
bcc>  exit
Router1>
```

Exiting the BCC returns you to the Technician Interface prompt.

If you need more detailed information about Technician Interface access, login, or logout procedures, refer to *Using Technician Interface Software*.

Using the BCC Configuration Hierarchy

The **help tree -all** and **show config -all** commands display the configuration hierarchy (or configuration tree) of any Bay Networks AN, ARN, ASN, BN, or System 5000 router. The **help tree -all** command displays the hierarchy of every object you *may* configure, but the **show config -all** command displays only the hierarchy of objects *actually* configured. The contents of the BCC configuration tree varies from device to device. (See "Terminology and Concepts" in Chapter 1 for more information about the BCC configuration hierarchy.)

Figure 2-1 illustrates a sample BCC configuration for an AN, BN, or ARN router.

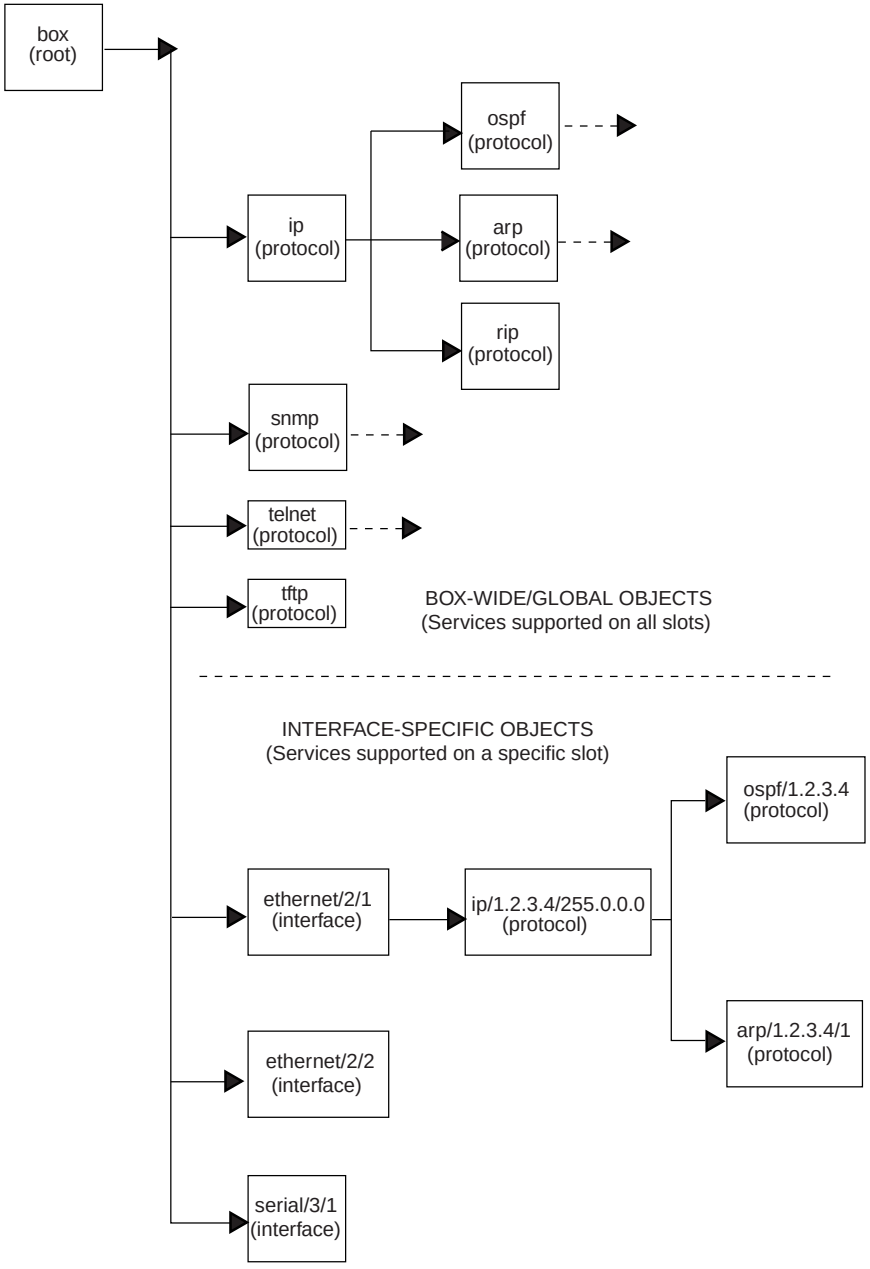

Figure 2-1. Sample BCC Configuration (BN Router)

Note the tree-like hierarchy of this configuration, where

- Box contains global objects IP, Telnet, TFTP, and SNMP, plus the physical interface objects ethernet/2/1, ethernet/2/2, and serial/3/1.
- Global IP contains the ARP, RIP, and OSPF global objects.

Locating Configured Objects

The BCC displays (and understands as command input) the location or *context* of any object such as an IP interface in terms of a path that begins at the root (box) level of the device configuration tree. For example, if you configure an IP interface (address 1.2.3.4, mask 255.0.0.0) on ethernet/2/1 of a BLN router, the BCC understands its location as *box; ethernet/2/1; ip/1.2.3.4/255.0.0.0* (Figure 2-2).

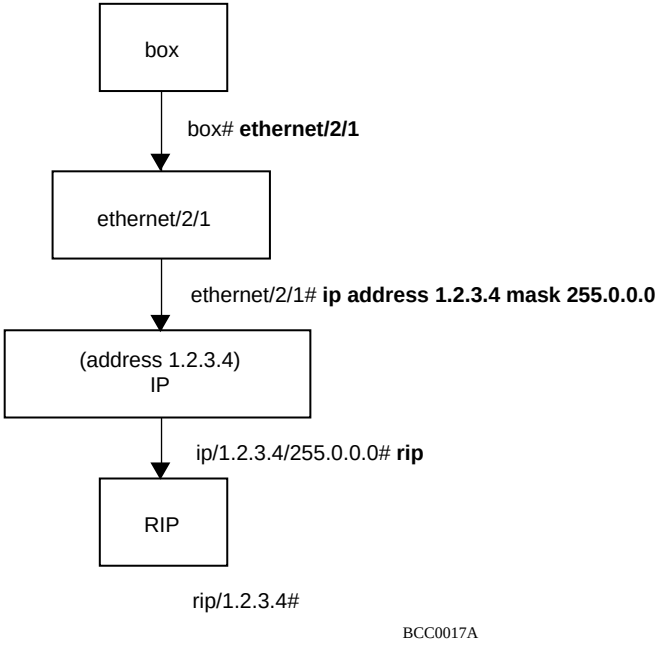

Figure 2-2. Object Location or Context

Each semicolon in the path marks a transition from one level to the next branch level in the device configuration tree. The semicolon is also equivalent to a Return key entered at the end of a command, effectively starting a new command line.

Displaying Context

In BCC configuration mode, use the **pwc** (print working context) command to display the path from root (box) level to your current level in the device configuration tree. For example:

```
rip/192.168.125.34# pwc
```

This command displays the full path from root (box) level to the current level, rip/192.168.125.34. The output appears as follows:

```
box; ethernet/2/1; ip/192.168.125.34/255.255.255.224; rip/192.168.125.34;
```

The **pwc** command displays the BCC instance identifier of each configured object in the path.

Enter the **pwc** command at any prompt in BCC configuration mode.

You can also make a quick determination of your current working location by simply looking at the configuration prompt, which is context-sensitive.

Context-Sensitive Prompts

The BCC shows in the command-line prompt your current working context (location within the BCC configuration tree).

Example: (BN router)

```
box# ethernet/2/1
ethernet/2/1# ip/192.168.125.34/255.255.255.224
ip/192.168.125.34/255.255.255.224# rip
rip/192.168.125.34#
```

Note: The prompt contains the BCC instance identifier of the object you specified in the previous command line.

The prompt does not show the *full* path to the current object from root level. The prompt shows only the object that terminates the entire path from root (box) level. To obtain the full path, use the **pwc** command.

Navigating in Configuration Mode

Using BCC configuration commands, you can

- Move back to a previous context or level.
- Move back to root (box) level.
- Move forward to the next level.
- Move from your current level to any other level in the device configuration tree.

You can navigate from one object to another in BCC configuration mode by using

- The **back** command
- Configuration commands

Navigating with the back Command

In BCC configuration mode, use the **back** command to move a specific number of levels back toward root level. Syntax for the **back** command is as follows:

back [<*n*>]

Entering the **back** command with no argument moves you back one level closer to root level.

Example: (BN router)

```
rip/192.168.125.34# back
ip/192.168.125.34/255.255.255.224# back
ethernet/2/1#
```

Entering the **back** command with an integer moves you from your current working location, back toward root, the number of levels you specify.

Example: (BN Router)

```
rip/192.168.125.34# back 2
ethernet/2/1#
```

In this example, the **back 2** command moves you from a current working location of rip/192.168.125.34, back two levels to ethernet/2/1 (with ip/192.168.125.34/255.255.255.224 as the intervening level).

Note: If you enter an integer value that exceeds the actual number of levels back to root (`box` or `stack`) level, the BCC returns to root level.

Navigating with Configuration Commands

Moving Back One or More Levels

To move from your current working level back one or more levels closer to root (box) level of the device configuration tree, enter the full BCC instance ID of the desired object, as follows:

Example (go back one level)

```
rip/192.168.125.34# ip/192.168.125.34/255.255.255.224
ip/192.168.125.34/255.255.255.224#
```

Example (go back two levels)

```
rip/192.168.155.151#  ethernet/2/1
ethernet/2/1#
```

In the second example, the BCC searches back toward root automatically until it finds a context or level where the object you specified (in this case, ethernet/2/1) exists in the router configuration tree. The BCC enters the context of this object, and the command-line prompt displays your new location.

Moving Back to Root Level

You can move back to root level in configuration mode by entering the name of the object at that level, as follows:

```
ip/1.2.3.4#  box (AN/ANH/ARN/BN router)
box#

ip/1.2.3.4#  stack (ASN or System 5000 router)
stack#
```

Moving Forward from Root (box) Level

To move from your current working level to the next configured level (Figure 2-3), enter the BCC instance ID of the desired object, as follows:

Example: (BN router)

```
box#  ethernet/2/1
ethernet/2/1#  ip/1.2.3.4/255.0.0.0
ip/1.2.3.4/255.0.0.0#  rip
rip/1.2.3.4#
```

Notice that a slash character (/) joins the name and any required parameter values to make a BCC instance ID for any configured object.

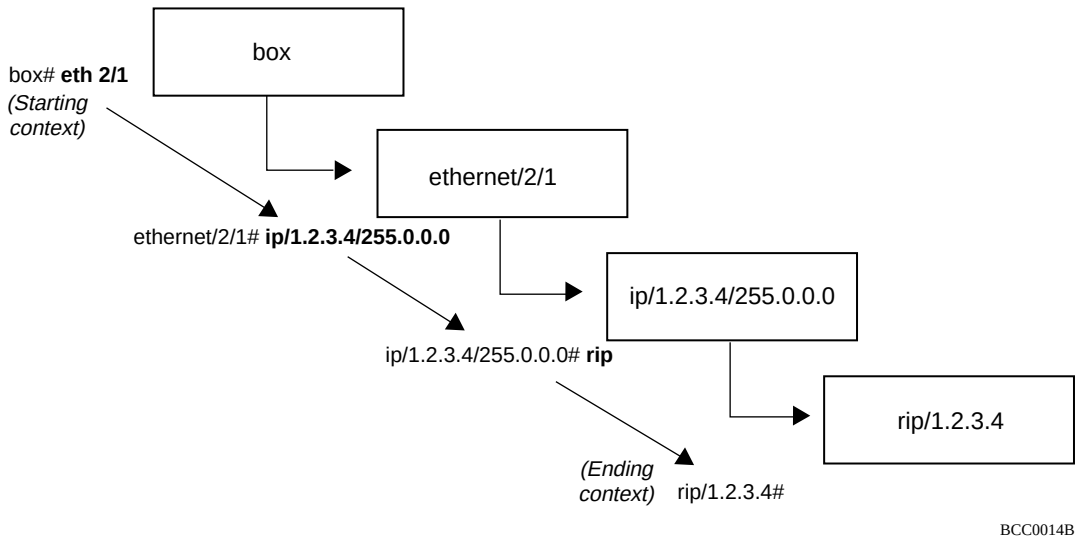

Figure 2-3. Moving Away from Root (box) Level

Specifying a Full Path

You can specify a full, or absolute, path from root (`box` or `stack`) level to any configured object. When you enter a path, specify the BCC instance identifier of each object.

Example: (BN router)

Move from ip/192.168.33.66/255.255.255.0 (on ethernet/2/1) to rip/1.2.3.4 on ethernet/2/2 (Figure 2-4).

```
ip/192.168.33.66/255.255.255.0# box;ethernet/2/2;ip/1.2.3.4/255.0.0.0;rip
 rip/1.2.3.4#
```

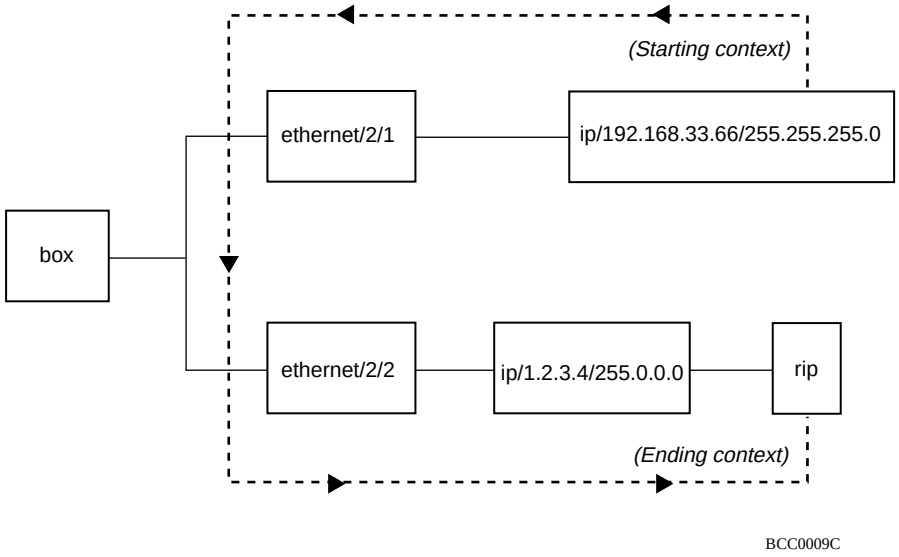

Figure 2-4. Specifying a Full Path

Specifying a Shortened Path

The BCC can automatically search backward (recursively), toward root level, until it finds a level where the object you specify first in the command line exists in the device configuration tree. After the BCC finds that object, it responds to any additional configuration (navigation) commands, starting from that level. This helps to shorten the command line necessary to navigate from one branch level to another.

Example: (BN router)

Move from ip/192.168.33.66/255.255.255.0 on ethernet/2/1 to rip/1.2.3.4 on ethernet/2/2 (Figure 2-5).

```
ip/192.168.33.66/255.255.255.0#  ethernet/2/2;ip/1.2.3.4/
255.0.0.0;rip
rip/1.2.3.4#
```

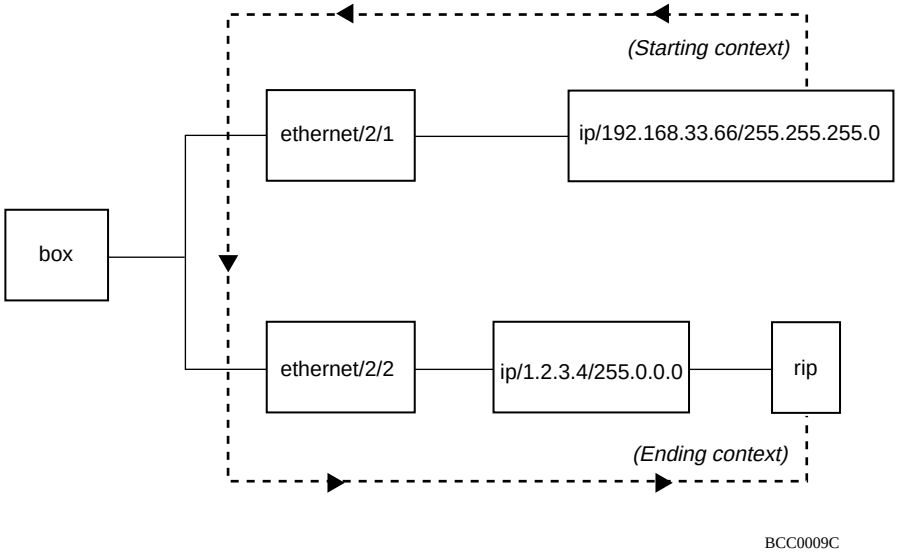

Figure 2-5. Navigating with the BCC Backward (Recursive) Search Feature

In this example, the BCC searches backward to find ethernet/2/2 (specified first in the command line), and then moves sequentially to the other locations (ip/1.2.3.4/255.0.0.0 and rip) specified next in the command line.

Displaying Online Help

The BCC supports the following online Help and Help-oriented features:

Help Feature	Command
Get Help on commands:	
Get an overview of Help-oriented command features.	**help**
List commands or command options you can enter.	**?** Or: *<command>* **?** Or: *<command> <option>* **?**
Display all BCC commands in alphabetical order (syntax). **Note**: Object and parameter names that you enter as configuration commands are not in this list. (Refer to the **?** and **help tree** commands later in this table.)	**help commands**
Display more detail on all BCC commands in alphabetical order (syntax and terse command descriptions).	**help commands -more**
Get full details on a specific command.	**help** *<command>* Example: To get full details on the **save** command, enter **help save**.
Get Help on symbols used to express BCC command syntax.	**help syntax**
Get configuration data (any BCC context or command mode):	
Show the device configuration tree, as follows: • List, in hierarchical format, all objects configurable in the device configuration tree, regardless of BCC mode (normal or configuration mode). • List, in hierarchical format, all objects configurable on the current object.	 **help tree -all** **help tree**
Show in BCC full syntax the configuration of the total device.	show config -all
Save the existing configuration to a file. Enter the **-all** option to save the total device configuration. Leave out the **-all** option to save only the configuration of the current object and its dependent subcontexts.	**show config[-all] -file [*<vol>*:] *<filename>***

Help Feature	Command
Get configuration data (BCC configuration mode only):	
List the names of objects, parameters, and commands you can enter next.	**?**
Get command usage Help and list parameters of an object you can add or modify from your current location.	*<object>* **?**
Get the current value for this parameter of the current object.	*<parameter>*
Get legal, current, and default values for this parameter of the current object.	*<parameter>* **?**

Get legal, current, and default values for this parameter of a configured object you can access from your current working context or level.	*<BCC_instance_ID> <parameter>* **?**
	Example: (BN router)
	`box# ethernet/2/1 bofl ?`
	`Current Value: enable`
	`Legal Value: {enable disable}`
	`Default Value: enable`

List values currently assigned to parameters of this object.	**info**
Describe parameters of the current object.	**help *<object_name>*** **help *<parameter_name>***
Describe parameters of an object outside the current context.	**help *<path> <object_name>*** (The *<path>* is the sequence of objects between root level and the desired context.) Example: **help ip ospf** Hint: Use the **help tree [-all]** command to determine the *<path>* to the desired object.
List other objects currently configured at the next (branch) level of the configuration tree for this device.	**lso**
Get Help on a specific task:	
Get Help on performing common BCC operations.	**help learning-bcc**
Get Help on how to edit BCC commands and command lines.	**help editing**

Example

Get a terse overview of BCC Help features:

```
bcc> help

                                Help Summary
     ____________________________________________________________________

System Help:

     * ?
     * <command> ?
          + Example: show ?
     * <command> <option> ?
          + Example: show ip ?
     * help
       Displays a summary of help-oriented commands.
     * help <any_command>
          + Examples:
            help pwc
            help show
     * help commands
       Displays a terse list of BCC system administrative commands.
     * help commands -more
       Displays more detail on BCC system administrative commands.
     * help editing
       Displays commands for editing command lines.
     * help tree
       Displays the hierarchy of all objects configurable from your
       current location.
     * help tree -all
       Displays, from any location, the hierarchy of all objects
       configurable on this device.
     * show config
       Displays the hierarchy of all objects already configured on ...
     * show config -all
       Displays the complete hierarchy of the actual/existing device
       configuration.
     * help syntax
       Defines symbols used to express BCC command syntax.
     * help learning-bcc
          .       .       .
          .       .       .
          .       .       .
```

Example

Get an alphabetical list of BCC system commands (syntax only, with some examples, where needed):

bcc> **help commands**

```
SCROLL TO THE COMMAND YOU NEED.
Note: This list does not include:
  * The names of objects or parameters you can enter as configuration
    commands
  * TCL (Tool Command Language) scripting commands supported by the
    Bay Command Console
  ____________________________________________________________________

                              Terse Help:

- - - - - - - - - - - - - - - - - - - - - - -

In any mode:

?

<command> ?

<command> <option> ?

- - - - - - - - - - - - - - - - - - - - - - -

In BCC configuration mode:

<object> ?
Example (from an ethernet context): ip ?

<BCC_instance_ID> ?
Example (from an ethernet context): ip/1.2.3.4/255.0.0.0 ?

<parameter> ?
Example: address-resolution ?

<BCC_instance_ID> <parameter> ?
Example (from an ethernet context):
ip/1.2.3.4/255.0.0.0 address-resolution ?
  ____________________________________________________________________

                              Commands:

      .                           .                           .
      .                           .                           .
      .                           .                           .
```

Example

Get a more detailed alphabetized list of BCC commands (with syntax and terse definitions):

```
box#  help commands -more

SCROLL TO THE COMMAND YOU NEED.
 Note: This list does not include:
   * The names of objects or parameters you can enter as configuration
     commands
   * TCL (Tool Command Language) scripting commands supported by the
     Bay Command Console
  ____________________________________________________________________

                              Terse Help:

 - - - - - - - - - - - - - - - - - - - - - - -

 (In any mode:)

 ?
        List the names of commands (in any mode), parameters of the
        current object (configuration mode only), and objects
        configurable from your current working location (configuration
        mode only).

 <command> ?
        <command> <option> ?
        Display (in any mode) usage help, keyword options, or -<xyz>
        filters for this command.

 - - - - - - - - - - - - - - - - - - - - - - -

 (In BCC configuration mode:)

 <object> ?

        List parameters of this branch object, which you can configure
        from your current location.
        Example (from an ethernet context): ip ?

 <BCC_instance_ID> ?

   .                              .                              .
   .                              .                              .
   .                              .                              .
  ____________________________________________________________________

                              Commands:
   .                              .                              .
   .                              .                              .
   .                              .                              .
```

Example

Define the purpose of each parameter of ethernet/2/1:

```
ethernet/2/1# help ethernet

Usage: "ethernet slot <value> connector <value>"
Or:    "ethernet <slot>/<connector>"

                            ethernet Parameters:
     ____________________________________________________________________


   bofl
          Allows breath-of-life polls to be disabled.

   bofl-retries
          Specifies the BOFL Retry Count.

   bofl-timeout
          Specifies the number of seconds for the BOFL timer.

   bofl-tmo-divisor
          Specifies the BOFL TMO divisor.

   circuit-name
          Specifies the circuit Name of this port.

   connector
          -REQUIRED- Specifies a connector on this interface.

   hardware-filter
          Enables the hardware bridge filter if available.

   receive-queue-length
          Specifies the number of receive buffers dedicated to the chip.

   slot
          -REQUIRED- Specifies the slot associated with this interface.

   state
          Indicates the administrative state of this object (enabled or
          disabled).

   transmit-queue-length
          Specifies the number of transmit buffers dedicated to the chip.

      .          .         .
      .          .         .
      .          .         .
```

Example (BN router)

Show all objects you can configure on this device:

```
box#  help tree -all

The entire configuration tree is:
  board
  virtual
    ip
      ospf
        neighbor
  ftp
  http
  ntp
    peer
  snmp
    community
      manager
    trap-entity
    trap-event
  tftp
  console
  telnet
    client
    server
  atm
    atm-interface
    signaling
      timers
      signaling-vc
      ilmi
        ilmi-vc
      sscop
    pvc-service
      pvc
      ip
        arp
        rip
        ospf
          neighbor
        rdisc
        igmp
          relay
      ipx
        rip
        sap
        static-route
        adjacent-host
        static-service
               .     .    .
               .     .    .
               .     .    .
```

Example (BN router)

List objects you can configure, starting at the ethernet/2/1 branch of the device configuration:

```
ethernet/2/1# help tree
The configuration tree below this context is:
  ip
    arp
    rip
    ospf
      neighbor
    rdisc
    igmp
      relay
  ipx
    rip
    sap
    static-route
    adjacent-host
    static-service
    route-filter
    server-network-filter
    server-name-filter
  auto-neg
```

Note: The main difference between the **help tree** and **show config** commands is that **help tree** shows what you *can configure*; **show config** shows you what *has been configured*. For more information about the **show config** command, see "Displaying the Configuration of One Object" on page page 2-25.

You can also use the output of the **help tree** [**-all**] command to find the configuration or navigation path to a specific object.

Example

List the names of all objects, parameters, and system commands you can enter from the context of ip/192.168.125.34/255.255.255.0:

```
ip/192.168.125.34/255.255.255.224# ?
Sub-Contexts:
  arp     igmp    ospf    rdisc   rip

Parameters in Current Context:
  address                 cost                    on
  address-resolution      end-station-support     proxy
  all-subnet-broadcast    has                     redirects
  assocaddr               host-cache-aging        state
  broadcast               mask                    udp-checksum
  cache-size              mask-reply
  configured-mac-address  mtu-discovery

System Commands:
  ?                       display                 ping
  back                    enable                  pktdump
  bccSource               exit                    pop
  bconfig                 format                  prom
  boot                    getcfg                  pwc
  cd                      help                    readexe
  clear                   help-file-version       record
  clearlog                history                 reset
  commit                  ifconfig                restart
  compact                 info                    rm
  config                  loadmap                 save
  cp                      log                     securelogin
  cwc                     logout                  show
  date                    lso                     stamp
  debug                   mget                    stop
  delete                  mlist                   system
  diags                   more                    tic
  dinfo                   mset                    type
  dir                     partition               unmount
  disable                 password                xmodem
```

The list of configurable objects (“Sub-Contexts”) are those you can add from your current location in the device configuration tree (Figure 2-6).

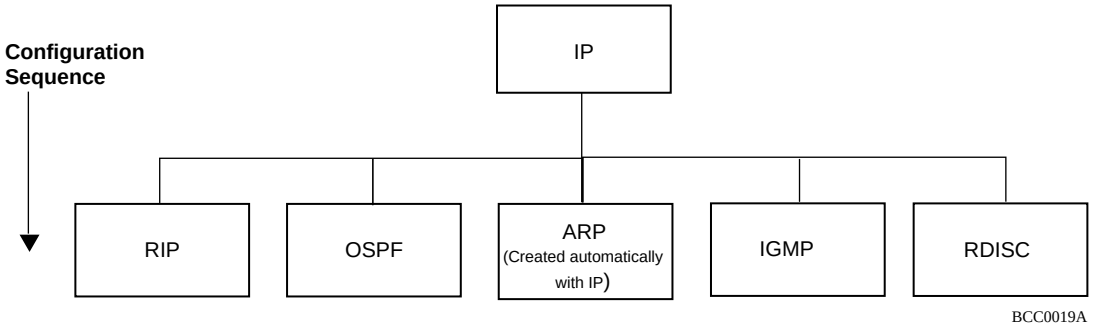

Figure 2-6. Objects to Configure at the Next (Branch) Level

Example: (BN router)

Get the values currently assigned to all parameters of IP (address 1.2.3.4) on ethernet/2/1:

```
ip/1.2.3.4/255.0.0.0# info
  on ethernet/2/1
  state enabled
  address 1.2.3.4
  mask 255.0.0.0
  assocaddr 0.0.0.0
  cost 1
  broadcast 1.2.3.6
  configured-mac-address 0x
  mtu-discovery off
  mask-reply off
  all-subnet-broadcast off
  address-resolution arp
  proxy off
  host-cache-aging cache-off
  udp-checksum on
  end-station-support off
  redirects on
  cache-size 128
```

To get the current value assigned to a specific parameter, enter just the parameter name, as follows:

```
ip/1.2.3.4/255.0.0.0#>  proxy
  proxy off
```

Using the BCC Help File

To use any BCC command that begins with the verb **help** (for example, **help commands**, or **help tree**), the BCC Help file (called *bcc.help*, by default) must be present on the NVFS default volume. If the file is missing, or if you have not configured the box to find the Help file, the BCC displays the following message:

```
The help data file is missing.  Please transfer "bcc.help" to the default
volume.  You can obtain a copy of "bcc.help" from

* Compact Disk (CD) for this software release. (Released version of the
file.)

* http://support.baynetworks.com/library/tpubs/bcc (Latest version of the
file.)

For more information on BCC help, refer to your Using the Bay Command
Console user guide.
```

If you see this message, first check to see if the Help file already exists on the default volume. If the file is missing, obtain a copy of it and use FTP or TFTP to transfer the file to the default volume of the device. After verifying that the file exists on the default volume, enter BCC configuration mode and enter the following command to specify the name and location of the Help file:

help-file-name [*<vol>* :] {**bcc.help** | *<file_name>* }

Examples: (BLN router)

```
box#  help-file-name 2:bcc.help
```
or:
```
box#  help-file-name 4:new.help
```

Note: If you choose a different name for the Help file, be sure to rename the file itself.

Displaying Help File Version Data

The BCC help-file-version command enables you to display version data for your BCC help file, *bcc.help*. Enter the command at any BCC prompt, as follows:

Example: (BN router)

```
box# help-file-version
Help file 2:bcc.help, contains this version data:
Data version is: 2.
Creation date is: 1997 Nov 20 14:42:40 hrs.
If a later version of the help file exists for this release, you can
use a web browser to obtain it from the following Bay URL:
   http://support.baynetworks.com/Library/tpubs/bcc
Follow the instructions at this website to obtain a copy of the BCC
help file you need.
```

Displaying Configuration Data

You can use the **show config** command to display Bay Networks device configuration commands and data.

Displaying the Total Device Configuration

The **show config -all** command displays the entire device configuration from any BCC prompt.

The output of **show config [-all]** describes

- Existing (configured) objects
- Objects that the BCC automatically added to the active device configuration
- Navigation (**back**) commands necessary to move to the context of the previous object configured, or to return to the root (box) level of the active device configuration

Example

```
bcc> show config -all
box type freln
  board slot 1
      type srml
  back
  board slot 2
      type qenf
  back
  board slot 3
      type wffddi2m
     .          .          .
     .          .          .
     .          .          .
  ethernet slot 2 connector 1
      circuit-name E21-alpha
    ip address 192.168.3.4 mask 255.255.255.224
        broadcast 192.168.3.5
      rip
      back
      arp
      back
    back
  back
     .          .          .
     .          .          .
     .          .          .
  ftp
      default-volume 2
  back
  snmp
    community label public
        access read-write
      manager address 0.0.0.0
      back
      manager address 192.168.9.9
     .          .          .
     .          .          .
     .          .          .
```

Note: Refer to the section “Identifying Board Types” in the *Router Software Release Notes* to translate board type values, such as “qenf” (see the entry for board slot 2, above), into detailed module descriptions, such as “Quad Ethernet without Hardware Filters” (“nf” = no filters).

Displaying the Configuration of One Object

You can view objects configured within a specific part of the BCC configuration tree by using the **show config** command or the **lso** command.

Example: (show config, BLN router)

You can use the **show config** command to view the configuration of a specific configured object. Navigate to the object in configuration mode, and enter the **show config** command. The following BLN router example shows everything configured on ethernet/2/1:

```
box# eth 2/1
ethernet/2/1# show config
ethernet slot 2 connector 1
    circuit-name E21-alpha
  ip address 192.168.125.34 mask 255.255.255.224
      broadcast 192.168.125.32
    arp
    back
    rip
    back
  back
back
```

Example (lso)

You can enter **lso** in configuration mode to view any instances of objects configured at your current location in the active device configuration, as follows:

```
ip/192.168.155.151/255.255.255.0#  lso
  arp/192.168.155.151/1   rip/192.168.155.151
```

Use the **lso** command if you have no need to display the contents of the current object in full BCC configuration syntax.

Displaying Binary Configuration Files as BCC Syntax

After booting the device from a binary configuration file, you can

1. Use the **show config** command to view the current device configuration in readable BCC syntax. For an example of **show config** output, refer to "Displaying the Total Device Configuration" on page page 2-23.
2. Enter new configuration commands to override elements of the active device configuration. (See Chapter 3 for more information about how to enter BCC configuration commands.)
3. Save the file as an ASCII configuration file that the BCC can read using the **source** command. (See "Saving Commands on a Workstation" below for instructions.)
4. Save the file as a binary configuration file, bootable on the same device or on another device. (See "Saving the Active Configuration as a Bootable Binary File" on page page 2-26 for instructions.)

Saving show config Output to an ASCII file on the Router

You can save the output of the show config command directly to a plain ASCII text file on the router, as follows:

To save the commands displayed by **show config** from any location in the device configuration tree:

show config -all -file *<volume>:<filename>*

To save only those commands displayed by **show config** (context-sensitive) from your current location in the device configuration tree:

show config -file *<volume>:<filename>*

Saving the Active Configuration as a Bootable Binary File

When you finish using BCC commands to modify an existing configuration, save the new configuration to a file on an NVFS (flash) volume. (At boot time, the router loses any configuration changes not previously saved to a NVFS volume.) The following command saves *config* as a bootable binary file on a volume you specify:

```
bcc>  save config <volume>:<filename>
```

Saving Commands on a Workstation

If you log in to a Bay Networks router from a workstation using Telnet or terminal emulation, you can use the native capabilities of the workstation to

- Save the output of any **show config** command from the screen to an ASCII file.
- Save a sequence of manually entered BCC commands to an ASCII file.

You can also use an ASCII text editor on a workstation to create a file from which the BCC can read configuration and system commands.

Note: If you create or edit ASCII files containing BCC commands, you must adhere to the BCC syntax requirements described in this guide. For example, you must include any commands necessary to navigate to each level of the device configuration tree, and any commands necessary to add, modify, or delete objects in the device configuration.

By saving BCC commands to an ASCII file, you can subsequently

- Edit the commands offline.
- Add comments to the file to describe details of the configuration.
- Save the edited file for later use.
- Use the BCC **source** command to edit the active configuration of Bay Networks devices of the same type and hardware configuration. (See “Entering Configuration Commands from a File” on page).

Chapter 3
Entering Commands

You can enter BCC configuration commands using basic (full), default, or abbreviated syntax. In addition, *command operators* enable you to perform certain operations more efficiently from your current location in the device configuration tree.

This chapter provides information about the following topics:

Topic	Page
Command Input Features	page 3-2
System Commands	page 3-6
Configuration Commands	page 3-7
Command Operators	page 3-11
Editing Command Lines	page 3-13

Command Input Features

This section contains information about

- Using Command Abbreviations
- Recalling Commands
- Entering Commands from a File
- Entering Multiple Commands per Line
- Continuing a Command Line (Underscore Prompt)
- Entering Comments

Using Command Abbreviations

For configuration command input, you can shorten existing object and parameter names (for example, eth = ethernet). You must enter a sufficient number of characters for the BCC to recognize that name uniquely. This is the BCC "minimum to distinguish" feature, which applies only when entering object or parameter names in configuration mode.

Example: (BN router)

```
box#  eth 5/1
ethernet/5/1#  back
box#  tf
tftp#
```

You can abbreviate system commands; for example, the BCC recognizes **sh** as **show** in contexts where there are no other command, configurable object, or parameter names that also start with **sh**.

Recalling Commands

The BCC supports a configurable command history buffer, from which you can recall commands recently entered. Recall/reenter commands from the history list as follows:

Purpose	Command
Recall the previous command from the history list.	Up arrow key or Control + p
Recall the next command from the history list.	Down arrow key or Control + n

The command history list contains up to 20 commands by default. You can increase the number of commands in the history list to a maximum of 40 by setting new values for the history parameter of the console and telnet objects.

Example

```
box# telnet
telnet# server
server# history 30
server# history
   history 30
server# back
telnet# back
box#
```

Entering Commands from a File

With Manager login privileges, you can use the **source** command in configuration mode to read BCC configuration and navigation commands from a designated ASCII source file into the active device configuration.

Caution: The **source** command makes immediate changes to the active device configuration.

The **source** command *merges* new configuration data from a file with existing data in device memory. If the file you specify contains configuration commands pertaining to objects already defined on the device, those commands dynamically overwrite the current configuration.

Note: If the BCC detects an error in the source file, it stops reading commands into the device configuration.

Syntax for the **source** command is as follows:

source *<volume>*:*<filename>*

For information about how to save as an ASCII file the output of the **show config** command, or a manually entered set of commands, refer to "Saving Commands on a Workstation" on page page 2-27.

Note: If you edit ASCII files containing BCC commands, you must adhere to BCC syntax requirements, including any commands necessary to navigate to each level of the device configuration tree.

Entering Multiple Commands per Line

To enter multiple commands in the same line, type a semicolon (;) wherever you would press Return to terminate a command.

Example: (BN router)

Configure ethernet/2/1 from root context, ip/1.2.3.4/255.0.0.0 on ethernet/2/1, and RIP on ip/1.2.3.4/255.0.0.0, as follows:

```
box#  ethernet/2/1;ip 1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

Continuing a Command Line (Underscore Prompt)

You can continue a command line by entering a backslash (\ ) character at the end of the current text line. The BCC treats characters on the next physical line as part of the same BCC logical command line.

You must immediately follow the initial escape character (\ ) with a newline (Return) character. The BCC treats these two characters and any trailing spaces as if they were exactly one space. Until you press Return without a preceding escape (\ ) character, the BCC replaces the # symbol in the context-sensitive prompt with an underscore (_) character.

Example

```
ip/1.2.3.4/255.0.0.0# cost 2 \
ip/1.2.3.4/255.0.0.0_ mask-reply on \
ip/1.2.3.4/255.0.0.0_ proxy on \
ip/1.2.3.4/255.0.0.0_ aging cache-on
ip/1.2.3.4/255.0.0.0#
```

Some command symbols normally used in pairs to denote the beginning and the end of a set of data also produce the continuation (underscore) prompt, including braces ({ }), brackets ([ ]), and quotes (“ ”).

Example:

```
box#  { ...
box_  ...}
box#
```

or:

```
box#  [...
box_  ...]
box#
```

or:

```
box#  “...
box_  ...”
box#
```

The BCC displays the underscore prompt after you type the opening symbol because it is expecting data plus the corresponding closing symbol.

Note: If you inadvertently type one of the opening symbols and see an underscore prompt, just type the corresponding closing symbol to restore the normal (#) prompt in BCC configuration mode.

Entering Comments

You can use a text editor (such as *vi* on a UNIX workstation) to add descriptive comments to a BCC configuration file. Enter comments in the following format:

```
box# <command> ;# comment
```

or

```
box# #comment
box# <command>
```

Example: (AN router)

```
box#  board slot 1 type andse;# 192.168.47.129  192.168.47.21
```

When you finish editing the file, save it on your workstation or PC. The comments are for reference only.

The BCC does not use or retain any comments you enter

- During a BCC configuration session with a Bay Networks device
- When importing a configuration file that contains comments (refer to "Saving Commands on a Workstation," in Chapter 2.)

For this reason, comments also do not appear in the output of any **show config** command.

System Commands

Bay Networks AN/ANH, ARN, ASN, BLN, and System 5000 routers support all system commands described in Appendix A. For help on a specific command, enter **help** <*command*> at any BCC prompt (for example, **help save**).

Configuration Commands

Caution: Configuration commands and **source** commands make real-time changes to the device configuration.

This section describes how to enter BCC configuration commands using any of the following formats:

- Basic (full) syntax
- Default syntax
- Abbreviated syntax

BCC syntax consists of object names, parameter names and values, and various types of punctuation. Note in particular that

- All object and parameter names appear as one word (hyphenated where necessary) in the BCC command line.
- Parameters have either a single value or multiple values enclosed in braces {x y z} in the command line. You either accept the default value or supply a value for each parameter associated with a configurable object.
- Parameters and their values must appear as a pair in the same command line.

Using Basic (Full) Syntax

The basic or full syntax for BCC commands consists of the following required and optional elements:

{*<object-name>*} {*<required_parameter> <value>* ... } ...
<parameter> <value> ... <parameter> <value>

The BCC requires input for any elements enclosed by braces ({ }).

{*<object-name>*} is the name of an object you want to configure (for example, **ip**)

The BCC assumes that an object you specify is new (and will create it) if it is not in the current configuration. If an object you specify already exists in the current configuration, the BCC assumes that you want to modify that object.

{<*required_parameter*> <*value*>} is any parameter-value pair required to uniquely identify an object you specify in a BCC command line. An object may have one or more required parameters.

For example, the full syntax for configuring an Ethernet interface on an AN/ANH, ARN, or BN router is

ethernet slot *<slot_no.>* **connector** *<connector_no.>*

`And on an ASN or System 5000 router, the full syntax is:`

ethernet slot *<slot_no.>* **module** *<module_no.>* **connector** *<connector_no.>*

Note: You cannot change the value of a parameter used by the BCC to create an instance identifier for any configured object. For example, you cannot modify the address value assigned to an IP interface previously configured. (Address is a required parameter for configuring an IP interface.) To change the value of any required parameter, you must delete the associated object, and then add it back into the device configuration with new required values.

[<*parameter*> <*value*>] is any parameter-value pair you can optionally customize for an object you specify in a BCC command line.

Using Default Syntax

Using default syntax, you do not have to enter the name of a required parameter; you enter only its value at the proper location in the command line.

Example (BN router)

ethernet *<slot>***/***<connector>*

With values:

ethernet 2/1

If you enter the name of an object without values for its required parameters, or with values inappropriate for its required parameters, the BCC returns usage Help, as shown in the following BN router example:

```
box# ethernet
Required parameter "slot" was not specified for ethernet.

Usage: "ethernet slot <value> connector <value>"
Or:    "ethernet <slot>/<connector>"
```

The following commands are equivalent.

Using full syntax:

```
box#  ethernet slot 2 connector 1
ethernet/2/1#
```

Using default (minimum) syntax:

```
box#  ethernet 2/1
ethernet/2/1#
```

ethernet is the object you want to configure.

slot and **connector** are the required parameters of **ethernet** on AN/ANH, ARN, and BN platforms.

The ability to configure objects using only the values for required parameters is the “minimum to configure” feature of the BCC. Typically, the BCC requires you to enter the values but not the names of required parameters. In some cases, however, the BCC requires you to explicitly enter the name of a required parameter along with its value. (The usage Help indicates the command elements you need to enter.)

Using Abbreviated Syntax

You can abbreviate BCC configuration commands, as follows:

Example: (BN router)

```
box# eth 2/1
```

This is the same as either of the following two commands:

```
box# ethernet slot 2 connector 1
```

```
box# ethernet 2/1
```

The BCC completes or expands abbreviated object and parameter names when you press Return.

If you press Return before entering a sufficient number of characters for the BCC to recognize the name of the object or parameter you want to configure, the BCC returns an error message:

Example: (BN router)

```
box# e

ambiguous command name "e": enable eof error ethernet eval exit expr
```

Note that the BCC returns a list of all the commands available in the current context, and that also start with the letter "e". Choose one command from the list, and enter enough characters for the BCC to recognize that command when you press Return.

You cannot abbreviate BCC instance identifiers.

Specifying Parameter Values

You must specify each parameter value in the form of a parameter-value pair. Each pair is a command argument pertaining to the object named first in the command line.

For example, the following command changes the BOFL timeout interval from 5 seconds to 4 seconds on ethernet/1/1 of a BN router:

```
box#  ethernet/1/1 bofl-timeout 4
```

ethernet is the object type, and **1/1** represents the specific slot and connector location for this Ethernet port.

bofl-timeout is the Ethernet interface parameter you want to modify, and **4** is the new value for the **bofl-timeout** parameter.

bofl-timeout 4 is the parameter-value pair.

Required, Derived, and Other Parameters

The BCC indicates when parameter values are required (you must supply a value) or derived (the BCC supplies a value). For all other parameters, the BCC supplies a user-modifiable default value.

Note: The BCC uses required parameters to make up the unique instance identifier for a configured object. You cannot change the value of any parameter that makes up the instance identifier for an object. To change these parameter values, you must delete the object, and then add it back into the device configuration with new values.

Specifying Multiple Parameter-Value Pairs

Within a specific context in configuration mode, you can

- Enter an object name and one parameter-value pair per command line.
- Enter an object name and multiple parameter-value pairs (each pair separated by a space) on the same command line.

Example (BN router; specifying one parameter-value pair per line)

```
box#  ethernet 2/1
ethernet 2/1#  bofl-retries 6
ethernet 2/1#  bofl-timeout 7
ethernet 2/1#  hardware-filter enabled
ethernet 2/1#
```

Example (BN router; specifying multiple parameter-value pairs per line)

```
ethernet 2/1#  ip address 1.2.3.4 mask 255.255.255.0 redirects off
ip.1.2.3.4/255.255.255.0#  ospf area 2.3.4.54 hello-interval 5
ospf/1.2.3.4#
```

Command Operators

BCC command operators perform a named operation within the current or specified configuration context. All Bay Networks AN/ANH, ARN, ASN, BN, and System 5000 routers support the following command operators in BCC configuration mode with read-write privileges (Table 3-1).

Table 3-1. BCC Command Operators

Operator	Function
disable	Change the administrative state of a configured object from enabled to disabled, as follows: `ip/1.2.3.4/255.0.0.0# disable` You can accomplish the same change by assigning the value “disabled” to the state parameter of an object you want to disable.
enable	Change the administrative state of a configured object from disabled to enabled, as follows: `ip/1.2.3.4/255.0.0.0# enable` You can accomplish the same change by assigning the value “enabled” to the state parameter of an object you want to reenable.
delete	Delete the object identified in the BCC context-sensitive prompt. For example, the following command deletes the IP interface (address 1.2.3.4): `ip/1.2.3.4/255.0.0.0# delete` **CAUTION:** Deleting an object at one level of the configuration tree causes the BCC to automatically delete any branches configured on that object. For example, if you delete an IP interface, the BCC deletes any protocols (such as RIP, ARP, or OSPF) configured on that interface.

info	List the names and values currently assigned to parameters of the current working context. For example, entering **info** after **ip** (at box level) lists the values currently assigned to parameters of the IP global object:
	`ip# info`
	`on box`
	`state enabled`
	`forwarding forwarding`
	`ttl 30`
	`cache-timeout default`
	`mib-table route`
	`all-subnets disabled`
	`classless disabled`
	`max-policies 32`
	`route-filters enabled`
	`rip-max-paths 1`
	`ecmp-method disabled`
	`isp-mode disabled`
	`ospf-max-paths 1`
	`icmp-error-limit 0`

Editing Command Lines

Table 3-2 describes the keystrokes you can use to edit BCC command lines.

Table 3-2. Keystrokes for Editing BCC Command Lines

Editing Function	Keystrokes
Move the cursor left	CONTROL + b or left arrow key
Move the cursor right	CONTROL + f or right arrow key
Delete the current line	CONTROL + u
Delete the word at the cursor location	CONTROL + w
Delete the character at the cursor location	CONTROL + d
Move the cursor to the beginning of the line	CONTROL + a
Move the cursor to the end of the line	CONTROL + e
Toggle insert mode	CONTROL + o
Delete previous character	BKSP or DEL, or CONTROL + h
Interrupt	CONTROL + c
Start echo to the screen	CONTROL + q
Stop echo to the screen	CONTROL + s
Recall previous command	CONTROL + p or up arrow key
Recall next command	CONTROL + n or down arrow key

For example, use the up arrow key (or CONTROL + p) to review your last input, and then use other control-key combinations to edit the command line as needed.

Chapter 4
Configuring a Network Device

This chapter describes how to complete the following tasks using BCC commands:

Topic	Page
Creating and Modifying a Device Configuration	page 4-1
Entering Configuration Commands from a File	page 4-15
Disabling a Configured Object	page 4-16
Enabling a Configured Object	page 4-17
Deleting a Configured Object	page 4-18
Configuration Command Responses	page 4-19

Creating and Modifying a Device Configuration

You configure a Bay Networks device by defining a set of objects that collectively define its behavior on a network. Each object has a set of parameters with values set either by you or by the device software. You configure a device starting at the root (box) level of the device configuration tree.

Following is a BCC configuration sequence that applies to a variety of network devices.

1. Open a Technician Interface session and start the BCC interface.
2. Start BCC configuration mode.
3. Use BCC configuration commands to create (add) new objects in the device configuration, and to modify default values for parameters of each object to meet the requirements of your network. Be sure to enable any box-wide protocols not enabled automatically by the BCC software, for example, TFTP and Telnet Server.

4. Use the **save** command to save your configuration as a bootable (binary) file on the device.
5. Exit BCC configuration mode, exit the BCC to the Technician Interface, and log out of the device.

You may find it helpful to first diagram what you want to configure in terms of the BCC configuration tree or hierarchy for the device. For example, refer to the following sample router configuration.

Sample Router Configuration

The following example shows a sequence of commands you can use to configure a BCN router on a network. Assumptions for this example are that you first complete the physical installation of the router, then boot the router using the image (*bn.exe*) and the minimum configuration file (*ti.cfg)*.

This example creates the following objects in the total router configuration (Figure 4-1):

- IP (global)
  - -- ARP (global) on IP
  - -- RIP (global) on IP
- SNMP (global)
  - -- Community “public” on SNMP
    - Manager (address 0.0.0.0) on Community “public”
- FTP (global)
- TFTP (global)
- Telnet (global)
  - -- Server (global) on Telnet
- Quad Ethernet interface in slot 13
  - -- IP interface (address 192.168.133.114) on Ethernet connector 1
    - ARP on IP interface 192.168.133.114
    - RIP on IP interface 192.168.133.114

- Serial interface in slot 5
- Dual Token Ring interface in slot 9
- FDDI interface in slot 11
- SRML system controller in slot 7

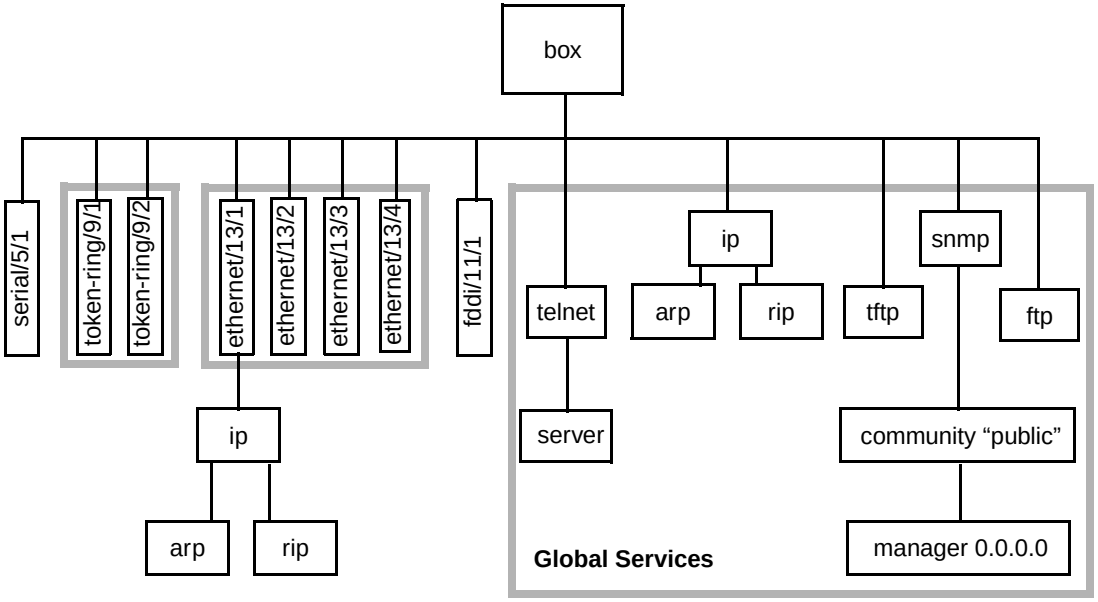

Figure 4-1. Sample BCC Configuration (BCN Router)

After you create a diagram of the device configuration tree, configure the device using a cycle of BCC configuration commands similar to those shown in Figure 4-2.

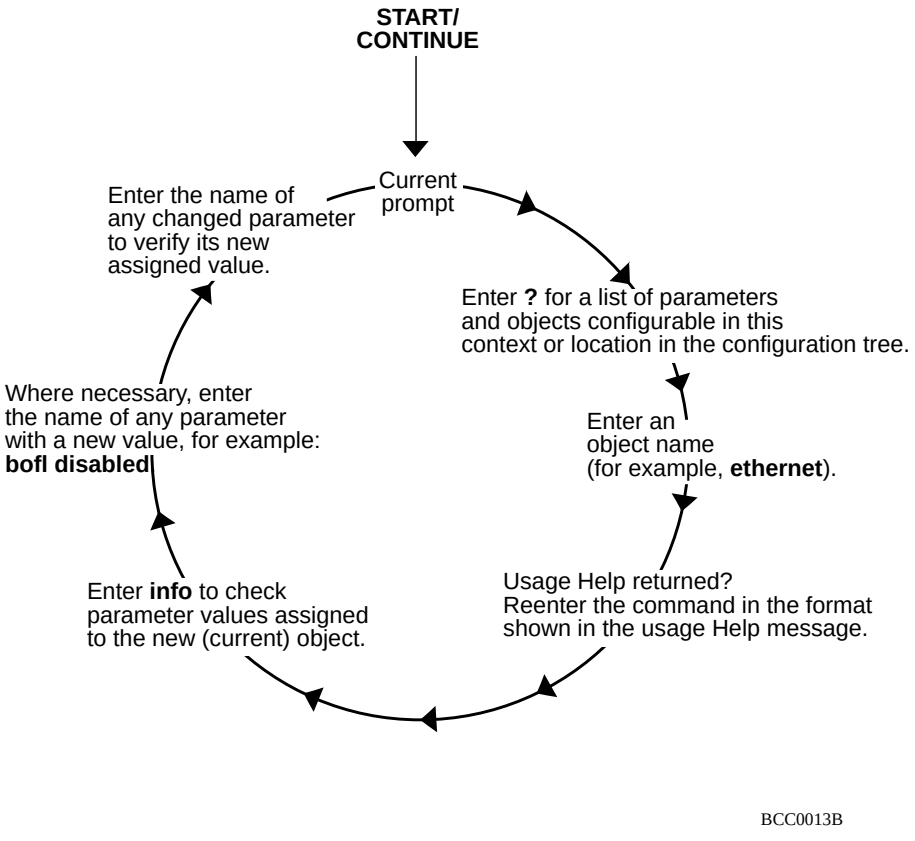

Figure 4-2. Typical BCC Configuration Cycle

To create the sample configuration shown in Figure 4-1 using BCC commands, proceed as follows:

1. **Log on to the router as Manager, so that you can modify the device configuration.**

```
Bay Networks, Inc. and its Licensors.
Copyright 1992, 1993, 1994, 1995, 1996, 1997. All rights reserved.

Login: Manager
Mounting new volume...
Device label:
Directory: 2:
New Present Working Directory: 2:

      Welcome to the Backbone Technician Interface

Router1>
```

2. **Start the BCC by entering bcc at the Technician Interface prompt.**

```
Router1> bcc

     Welcome to Bay Command Console!

     * To enter configuration mode, type config

     * To list all system commands, type ?

     * To exit the BCC, type exit

bcc>
```

3. **Enter BCC configuration mode by entering config.**

```
bcc>  config
box#
```

4. **Check the hardware configuration of the router.**

```
box# show config

box type frecn (BCN router with a FRE-2 controller)
  board slot 5
      type sync (Serial link module in slot 5)
  back
  board slot 7
      type srml (System resource module in slot 7)
  back
  board slot 9
      type dtok (Dual token ring link module in slot 9)
  back
  board slot 11
      type wffddi2m (Multimode FDDI link module in slot 11)
  back
  board slot 13
      type qenf (Quad Ethernet with filters in slot 13)
  back
  console portnum 1 (console device on port 1)
      prompt {"%slot%:"}
      auto-manager-script automgr.bat
      auto-user-script autouser.bat
  back
back
box#
```

Note how output of the **show config** command automatically includes navigation (**back**) commands. If you save this output to a file, you can reenter them automatically using the BCC **source** command. (See "Entering Commands from a File" in Chapter 3.)

For information on how to determine module types based on the assigned value of the board "type" parameter, refer to the section, "Identifying Board Types" in the BCC section of the *Router Software Release Notes.*

5. **Choose a port (interface type, slot, and connector) for the initial IP interface to the router.**

```
box# ethernet slot 13 connector 1
ethernet/13/1#
```

6. Check to see what you can configure (sub-contexts and parameters) at this level.

```
ethernet/13/1# ?
Sub-Contexts:
  ip    ipx

Parameters in Current Context:
  bofl                    circuit-name            on
  bofl-number             connector               receive-queue-length
  bofl-retries            hardware-filter         slot
  bofl-timeout            has                     state
  bofl-tmo-divisor        name                    transmit-queue-length

System Commands:
  ?               delete          mkdir           reset
  attribute       diags           mlist           restart
  back            dinfo           more            rm
  boot            dir             mount           rmdir
  cd              disable         mset            save
  clear           enable          partition       securelogin
  clearlog        exit            password        show
  commit          format          ping            source
  compact         help            pktdump         stamp
  config          history         pop             stop
  context         loadmap         prom            system
  cp              log             pwc             tic
  cwc             logout          readexe         unmount
  date            lso             reconfig        xmodem
  debug           mget            record
ethernet/13/1#
```

Based on the “Sub-Contexts” list, you can add **ip** (interface IP) or ipx (interface IPX) to this Ethernet port. At this level, the list of configurable objects depends on the *board* type described in the current prompt. For example, 100BASE-T modules additionally allow you to configure an “autoneg” (autonegotiation) object at this level.

You can also configure (modify) values currently assigned to “Parameters in Current Context” (parameters of ethernet/13/1), shown above.

The list of “System Commands” always appears along with information about configurable objects and parameters.

7. Add IP (address 192.168.133.114) to ethernet/13/1.

```
ethernet/13/1# ip 192.168.133.114
Usage: "ip address <value> mask <value>"
Or:    "ip <address>/<mask>"
Required parameter "mask" was not specified for ip.
```

The error message appears because the BCC requires you to enter a mask value whenever you create an IP interface. Because the first octet of the address is 192 (0xC0), this is a Class C address, requiring the first 3 octets to be the network portion of the interface address. You can express the corresponding mask value as either 255.255.255.0 (using dotted-decimal notation) or 24 (the number of bits making up the network portion of the IP interface address), as follows:

```
ethernet/13/1# ip 192.168.133.114/24
ip/192.168.133.114/255.255.255.0#
```

The BCC converts the integer representing the number of mask bits into a dotted-decimal mask value, as shown in the last prompt.

8. Check the values currently assigned to parameters of IP on this interface.

```
ip/192.168.133.114/255.255.255.0# info
  state enabled
  address 192.168.133.114
  mask 255.255.255.0
  assocaddr 0.0.0.0
  cost 1
  broadcast 0.0.0.0
  configured-mac-address 0x
  mtu-discovery off
  mask-reply off
  all-subnet-broadcast off
  address-resolution arp
  proxy off
  host-cache-aging cache-off
  udp-checksum on
  end-station-support off
  redirects on
  cache-size 128
```

9. Change the subnet mask to 255.255.255.224.

```
ip/192.168.133.114/255.255.255.0# mask 255.255.255.224
"mask" parameter may not be modified
ip/192.168.133.114/255.255.255.0#
```

You cannot modify any parameter values included in the BCC instance identifier of a configured object, in this case, ip/192.168.133.114/255.255.255.0. To change the value of any parameter that is part of a BCC instance ID, you must first delete the object, then re-create it with the desired parameter values. In this case, you must delete the IP interface and then re-create it on ethernet/13/1, using the mask value of 255.255.255.224, as follows:

```
ip/192.168.133.114/255.255.255.0# delete
ethernet/13/1# ip 192.168.133.114/255.255.255.224
ip/192.168.133.114/255.255.255.224#
```

10. Check to see what you can configure at this level.

```
ip/192.168.133.114/255.255.255.224# ?
Sub-Contexts:
  arp     igmp    ospf    rdisc   rip

Parameters in Current Context:
  address                 cost                    name
  address-resolution      end-station-support     on
  all-subnet-broadcast    has                     proxy
  assocaddr               host-cache-aging        redirects
  broadcast               mask                    state
  cache-size              mask-reply              udp-checksum
  configured-mac-address  mtu-discovery

System Commands:
  .       .       .
  .       .       .
  .       .       .
```

You can modify values currently assigned to parameters of ip/192.168.133.114/255.255.255.224, or you can add RIP, OSPF, Router Discovery, IGMP, or ARP to this interface.

11. Add RIP as the routing protocol (by default, RIP1) on this interface.

```
ip/192.168.133.114/255.255.255.224# rip
rip/192.168.133.114#
```

12. Return to root (box) level to configure global system services.

```
rip/192.168.133.114# box
box#
```

13. Check to see what global services (protocols) and interfaces you can configure at this level.

```
box# ?
Sub-Contexts:
  atm             fddi            ipx             serial          virtual
  backup-pool     ftp             isdn-switch     snmp
  board           hssi            mce1            telnet
  console         http            mct1            tftp
  ethernet        ip              ntp             token-ring

Parameters in Current Context:
  build-date      description     location        system-name
  build-version   has             mib-counters    type
  contact         help-file-name  on              uptime

System Commands:
.       .       .
.       .       .
.         .       .
```

You can add any of the following global services (affecting all slots): IP, IPX, FTP, NTP, SNMP, Telnet, and TFTP.

You can add any of the following interfaces: ATM, Ethernet, FDDI, HSSI, Serial, Token-ring, or Virtual.

You can view the configuration of a board in any slot, but you cannot modify the parameters of any board object.

14. List the objects already configured at box level.

```
box# lso
board/11        board/5         board/9         ethernet/13/1
board/13        board/7         console/1       ip
box#
```

When you added the first instance of IP to the box (ip/192.168.133.114/255.255.255.224), the BCC automatically created the global IP object at box level.

15. Add SNMP to the device.

```
box# snmp
snmp#
```

16. Check to see what you can configure next at this level.

```
snmp# ?
Sub-Contexts:
  community       trap-entity     trap-event

Parameters in Current Context:
  authentication-traps    lock-timeout            state
  has                     name                    type-of-service
  lock                    on
  lock-address            scope-delimiter

System Commands:
.       .       .
.       .       .
.         .       .
```

You can modify values currently assigned to attributes of SNMP, and you can add a community, define a trap entity, or define a trap event.

17. View the parameter definitions available using the BCC help parameters command.

```
snmp# help snmp
```

```
                              snmp Parameters:
   ____________________________________________________________________

   state
          Indicates the administrative state of this object (enabled or
          disabled).

   authentication-traps
          Sends trap for sets from false Mgr or Community.

   lock
          Allows the locking mechanism to be disabled.

   lock-address
          Allows the lock address to be cleared.

   lock-timeout
        .           .           .
        .           .           .
        .           .           .
```

18. Define the SNMP community "public."

```
snmp# community public
community/public#
```

19. Check the values currently assigned to parameters of this SNMP community.

```
community/public# info
  on snmp
  label public
  access read-only
  scope-type {}
```

20. To allow network management applications (such as Site Manager) to modify the device configuration, modify the value of the access attribute to read-write.

```
community/public# access read-write
community/public#
```

21. Define an SNMP manager for the router.

```
community/public# manager
Usage: "manager address <value>"
Or:    "manager <address>"
Required parameter "address" was not specified for manager.
```

The BCC error message indicates what you left out and automatically provides extended usage Help on how to configure an SNMP manager.

22. Try again to add the manager, this time supplying a value for its required attribute (parameter), address. (You must enter a value but not the name for a required parameter.)

```
community/public# manager 0.0.0.0
manager/public/0.0.0.0#
```

23. Enable the Telnet server entity on the router.

```
manager/public/0.0.0.0# telnet
telnet#  server
server#
```

24. Add TFTP services globally to the router.

```
server# tftp
tftp#
```

The BCC automatically searches back (toward root) to find the parent context suitable for TFTP (in this case, box). The BCC then adds TFTP to the device configuration. Note the new (`tftp#`) prompt.

25. Check values currently assigned to attributes of TFTP.

```
tftp# info
  on box
  state enabled
  default-volume 2
  retry-timeout 5
  close-timeout 25
  retry-count 5
```

26. Change the default volume number for TFTP to 5.

```
tftp# def 5
tftp#
```

27. Verify the change to the default-volume number.

```
tftp# def
  default-volume 5
```

28. Add FTP globally to the router.

```
tftp# ftp
ftp#
```

The BCC automatically searches back (toward root) to find the parent context suitable for FTP (in this case, box). The BCC then adds FTP to the device configuration. Note the new (ftp#) prompt.

29. Check the definitions for parameters of FTP.

```
ftp# help ftp
                                ftp Parameters:

_____________________________________________________________________

   on
          Identifies the parent(s) of this object.

   state
         Indicates the administrative state of this object (enabled
or
          disabled).

   default-volume
          Specifies the default volume where transferred files are
          written/retrieved.
     .       .       .
     .       .       .
     .       .       .
```

30. Check values currently assigned to parameters of FTP.

```
ftp# info
  on box
  state enabled
  default-volume 2
  login-retries 3
  idle-timeout 900
  max-sessions 3
  tcp-window-size 60000
```

31. Change the default volume number to 5.

```
ftp# def 5
ftp#
```

32. Verify the change to the default volume number.

```
ftp# def
  default-volume 5
```

33. Recheck the total device configuration.

```
ftp# show config -all
box type frecn
  board slot 5
      type sync
  back
  board slot 7
      type srml
  back
  board slot 9
      type dtok
  back
  board slot 11
      type wffddi2m
  back
  board slot 13
      type qenf
  back
  console portnum 1
      prompt {"%slot%:"}
      auto-manager-script automgr.bat
      auto-user-script autouser.bat
  back
  ethernet slot 13 connector 1
      circuit-name E131
    ip address 192.168.133.114 mask 255.255.255.224
     .       .       .
     .       .       .  (remaining configuration not shown here)
```

34. **Return to root level.**

```
ftp# box
box#
```

35. **Save the file using a name other than *config* until you can test the configuration.**

```
box#  save config startup.cfg
```

36. **Test the intial IP interface.**

```
box#  ping 192.168.133.114
IP ping: 192.168.133.114 is alive (size = 16 bytes)
```

37. **Ensure that the initial IP interface connects to another device on the network.**

```
box# ping 192.168.133.97
IP ping: 192.168.133.97 is alive (size = 16 bytes)
```

38. **When you finish configuring the router, exit configuration mode.**

```
box#  exit
bcc>
```

39. **Exit the BCC, which returns you to the Technician Interface prompt.**

```
bcc>  exit
Router1>
```

40. **Enter the logout command to close your console or Telnet session with the router.**

```
Router1> logout
```

Entering Configuration Commands from a File

If you have Manager (read-write) login privileges, you can use the **source** command to read BCC configuration and navigation commands from a designated ASCII source file into the active device configuration.

Caution: The **source** command makes immediate changes to the active device configuration.

The **source** command takes new BCC configuration commands and data from an ASCII file and *merges* those commands with existing configuration data in active device memory. If the file you specify in the **source** command contains configuration commands pertaining to objects already defined on a device, the commands in the file dynamically overwrite the configuration of those objects.

Syntax for the **source** command is as follows:

source *<volume>*:*<filename>*

For information about how to save as an ASCII file the output of the **show config** command or a manually entered set of commands, refer to "Saving Commands on a Workstation" in Chapter 2.

Disabling a Configured Object

In most cases, the BCC automatically enables objects that you add to the device configuration. However, you may need to disable an object to manage or troubleshoot the device. Here is a BN router example of how to disable an object (**rip**) on **ip/1.2.3.4/255.0.0.0**:

1. **Specify the configuration context for the object you want to disable.**

```
box#  ethernet/2/1;ip/1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

2. **Disable RIP.**

```
rip/1.2.3.4#  disable
rip/1.2.3.4#
```

3. **Verify that you disabled RIP.**

```
rip/1.2.3.4#  state
  state disabled
rip/1.2.3.4#
```

You can also disable an object from its parent context, using the following syntax:

disable *<BCC_instance_identifier>*

Example

```
ip/1.2.3.4/255.0.0.0#  disable rip/1.2.3.4
ip/1.2.3.4/255.0.0.0#
```

Using this method, you remain in the current context after disabling the branch object.

Enabling a Configured Object

If you disable a configured object to manage or troubleshoot the local device, you can use the BCC **enable** command to reenable that object. Here is a BN router example of how to enable an object (**rip**) previously disabled on **ip/1.2.3.4/255.0.0.0**:

1. **Specify the configuration context for RIP.**

```
box#  ethernet/2/1;ip/1.2.3.4/255.0.0.0;rip
rip/1.2.3.4#
```

2. **Reenable RIP.**

```
rip/1.2.3.4#  enable
rip/1.2.3.4#
```

3. **Verify that you reenabled RIP.**

```
rip/1.2.3.4#  state
  state enabled
rip/1.2.3.4#
```

You can also enable an object from its parent context, using the following syntax:

enable *<BCC_instance_identifier>*

Example

```
ip/1.2.3.4/255.0.0.0#  enable rip/1.2.3.4
ip/1.2.3.4/255.0.0.0#
```

Using this method, you remain in the current context after enabling the branch object.

Deleting a Configured Object

Because of the tree hierarchy, objects on higher branches of the tree depend on the state (and existence) of objects closer to the root of the tree. Deleting an object also deletes anything configured on that object.

Here is a BN router example of how to delete an IP interface from the active device configuration:

1. **Navigate to the object you want to delete:**

```
box#  ethernet/13/1
ethernet/13/1#  ip/192.168.133.114/255.255.255.224
ip/192.168.133.114/255.255.255.224#
```

2. **List all objects configured on the current object:**

```
ip/192.168.133.114/255.255.255.224# lso
arp/192.168.133.114/1   rip/192.168.133.114
```

3. **Delete the object.**

```
ip/192.168.133.114/255.255.255.224#  delete
ethernet/13/1#
```

4. **Verify that you deleted the object.**

```
ethernet/13/1# lso
```
(no objects listed)

Note that ip/192.168.133.114/255.255.255.224 no longer appears in the list of objects configured on ethernet/13/1. With a single **delete** command, the BCC automatically deleted the branch objects (arp/192.168.133.114/1 and rip/192.168.133.114) configured on ip/192.168.133.114/255.255.255.224.

You can also delete an object from its parent context, using the following syntax:

delete *<BCC_instance_identifier>*

Example

```
ip/1.2.3.4/255.0.0.0#  delete rip/1.2.3.4
ip/1.2.3.4/255.0.0.0#
```

Using this method, you remain in the current context after deleting the branch object.

Configuration Command Responses

The BCC completes the configuration task you entered unless a syntax, semantics, or completion error occurs. You can display and verify the configuration by entering the **show config** command.

In certain circumstances, the BCC may be unable to complete a configuration command; this is a completion error. When this occurs, the BCC displays an appropriate error message.

Appendix A
System Commands

AN/ANH, ARN, ASN, BCN, BLN, and System 5000 routers support all system commands shown in Table A-1. Note that some commands perform functions specific to your current interface level (Technician Interface, BCC, or BCC configuration mode) and login privilege level (Manager or User).

Table A-1. System Commands

Command	Purpose	Login/Access Privileges	
		Manager	User
?	List the names of objects, parameters, and system commands you can enter next.	✔	✔
<command> **?**	Display syntax usage help for *<command>.*		
<object> **?**	List parameters of an object you can access or configure from your current level in the device configuration tree.		
<parameter> **?**	List the current, default, and legal values for this parameter of the current object.		
<object> <parameter> **?**	List the current, default, and legal values for the specified parameter of this object.		
! [*<n>*] (Technician Interface only)	Repeat the last command, or repeat the last command *<n>* times.	✔	✔

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
back [*<n>*] (BCC configuration mode only)	Move your current working location or level back *<n>* levels closer to the root level of the BCC configuration tree.	✔	✔
bcc	Start the BCC from the Technician Interface prompt.	✔	✔
boot [{*<vol>*:*<image_name>* | - } { *<vol>*: *<config_name>* | - }]	Reboot the system.	✔	
cd [*<vol>*:][*<directory>*]	Set or display the current working volume and directory.	✔	✔
clear *<subcommands> <flags>*	Clear specific device information, such as IP and RIP data.	✔	
clearlog [*<slot_ID>*]	Clear all previous events from the system event log.	✔	
commit	Commit (make effective) new values that you assign to MIB attributes.	✔	
compact *<volume>*:	Compact existing files into a contiguous address space on a volume, and compact all unused space into a single contiguous block of free space for new files on the same volume.	✔	

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
config [**-read-write**|**-read-only**] (BCC only)	Enter BCC configuration mode. If you logged on as Manager, you have read-write access to the device configuration; if you logged on as User, you have read-only access, which allows you to "show" or navigate the existing configuration, but prevents you from changing the configuration. If you enter BCC configuration mode as Manager and want to change your privilege level for the current session from read-write to read-only, enter **config -read-only**. To change Manager privileges back to read-write, enter **config -read-write**.	✔	✔
copy *<vol>:<filename1> <vol>:<filename2>* (Technician Interface only)	Copy the contents of the first file to the second file.	✔	
cp *<vol>:<filename1> <vol>:<filename2>* (BCC only)	Copy the contents of the first file to the second file.	✔	
date [*<mm/dd/yy>*] [*<hh:mm:ss>*] [*<+|-><hh:mm>*]	Display or update the system time and time zone.	✔	✔
delete *<filename> <volume>* (Technician Interface only; see **rm** for BCC equivalent)	Delete the file from the specified volume.	✔	
delete [*<BCC_instance_ID>*] (BCC configuration read-write mode only)	Delete the object described in the current prompt, or delete the object known by the specified *<ID>* and configured previously from the current level.	✔	

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
diags [*<slot _ID>*]	Perform CPU/memory, backbone, and link diagnostics before downloading the GAME image and rebooting on the specified slot.	✔	
dinfo	Display the status of each file system volume on this device.	✔	✔
dir [*<vol>***:**]	Display the contents of the file system volume specified.	✔	✔
disable (BCC configuration read-write mode only)	Disable the object at your current location in the BCC configuration tree.	✔	
enable (BCC configuration read-write mode only)	Enable the object at your current location in the BCC configuration tree.	✔	
format *<volume>*:	Erase the entire contents of the file system volume specified and reinitialize it to a usable state.	✔	
get {*<obj_name>*|*<obj_id>*}**.**{*<attr_name>*|*<attr_id>*|*}[.{*<inst_id>*|*}] (Technician Interface only)	Retrieve the values of data objects in the MIB.	✔	✔
getcfg (Technician Interface, AN routers only)	Display network boot parameters. (Display the current parameter settings used to determine the source of image and configuration files.)	✔	✔

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
help [[**commands**] [**-more**] | *<command>* | **syntax** | **tree** [*<box_level_object>*] | **learning-bcc** | **editing**] (BCC only)	The word **help**, entered without arguments, displays an overview of BCC Help-oriented features. That screen describes how to get	✔	✔
help commands	• An alphabetical list of all commands, with syntax and terse descriptions		
help *<command>*	• Detailed Help on a specific command		
help syntax	• Help on how to interpret BCC syntax symbols		
help tree [-all]	• A hierarchical list (the configuration tree) of all objects you can configure * On this platform (use **-all**) * On the current object (omit **-all**)		
help *<object_name>*	• Definitions for parameters of the object described by its path from box level (AN/ARN/BN) or stack level (ASN/System5000). For example, **help ip ospf** defines parameters of the global ospf object, and **help ethernet ip** defines parameters of the IP interface on an Ethernet port.		
help *<parameter_name>*	• Definitions of all parameters of the current object. For example, **help bofl** causes the BCC to display the definitions of all parameters of ethernet. (**help** *<parameter_name>* is context-sensitive.)		
help [*<task>*]	• Help on how to perform a specific task. The tasks (and their corresponding **help** commands) are **help learning-bcc** **help editing**		

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
help [*<command>* | **-all**] (Technician Interface only)	The **help** command, entered without arguments at the Technician Interface prompt, displays an alphabetical list of all commands with syntax only. The list excludes commands available only in BCC mode. The combination **help** *<command>* displays detailed Help on a specific command, excluding any commands available only in BCC mode. The combination **help -all** displays, in scrollable format, detailed Help on all system commands, excluding those available only in BCC mode.	✔	✔
history [*<n>*] (	Display the command history list or recall command number *<n>* from the history list.	✔	✔
info (BCC configuration mode only)	List values currently assigned to all configurable parameters of this object.	✔	✔
ip *<subcommand> <flags>* (Technician Interface only)	Display data from IP; show a different view for each subcommand or flag you enter.	✔	✔
ip6 *<subcommand> <flags>* (Technician Interface only)	Display data from IPv6; show a different view for each subcommand or flag you enter.	✔	✔
list [[*<instances>* [*<obj_name>*]]] (Technician Interface only)	List objects in the MIB.	✔	✔
loadmap [*<slot_list>* | **all**] [*<filepath>*]	Display the load address and size of each dynamically loadable application (for example, a protocol).	✔	

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
log [*<vol>*:*<logfile>*] [**-d***<date>*] [**-t***<time>*] [**-e"***<entity>***"**] [**-f***<severity>*] [**-s***<slot_ID>*] [**-p**[*<rate>*] [**-c***<code #>*]	Display the current system event log. Follow the optional **-e** flag immediately with the entity name in quotes and uppercase characters (no intervening spaces). Specify *<severity>* using letters and no intervening spaces: f = fault w = warning i = info t = trace d = debug **Examples:** -fwid -ffitd -fwi -fwitd Use the optional **-p** flag to set an interval for polling the log and displaying the result.	✔	✔
log [**-x**|**-i**] [**-e"***<entity>"*] [**-f***<severity>*] [**-s***<slot_ID>*]	Excludes (**-x**) or includes (**-i**) event logging indicated by the command options.	✔	✔
log -z [**-s***<slot_ID>*]	Displays current filter setting.	✔	✔
logout	Exit the current login session.	✔	✔
lso [**-l**] (BCC configuration mode only)	List next-level (branch) objects configured on the current object. The optional **-l** flag causes the BCC to list object IDs in wrap-around screen format.	✔	✔
mget {*<obj_name>* | *<obj_id>*} . {*<attr_name>* | *<attr_id>* | * } [. {*<inst_id>*|*}] (BCC only)	Retrieve the values of data objects in the MIB.	✔	✔
mlist [[*<instances>* [*<obj_name>*]]] (BCC only)	List objects in the MIB.	✔	✔
more [**on** | **off**] [*<#_lines_per_screen>*]	Set or display the status of the **more** utility.	✔	✔

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
mset {*<obj_name>*|*<obj_id>*}**.**{*<attr_name>*| *<attr_id>*}.{*<inst_id>*} {*<value>*} (BCC only)	Modify (set) the values of data objects in the device MIB.	✔	
partition {create|delete} [*<vol>***:**]	Create or delete a partition on existing file system media.	✔	✔
password Manager	Change the password of the Manager account.	✔	
password User	Change the password of the User account.	✔	✔
ping -*<protocol> <address>* [**-t***<timeout>*]	Initiate an ECHO request/reply handshake.	✔	✔
pktdump *<line_number>* [**-s***<start>*] [**-c***<count>*]	Display packets that have been captured by an interface configured for Packet Capture.	✔	
prom [**-v**|**-w**] *<vol>*:*<ROM_Update_File><slot_ID>* [*<slot_ID>*...]	Update or verify the software located on a flash PROM device.	✔	
pwc (BCC configuration mode only)	Display the path to your current working location or level in the tree, starting from root level.	✔	✔
readexe *<vol>*:*<filename>*	Validate the checksums of an executable image and print out all the file header information.	✔	✔
record {**open**|**close**} [**-fileonly**] [**-pause**] *<vol>*:*<filename>* (Technician Interface only)	Record to a file all messages written to the terminal. You can open, pause, and close a recording session.	✔	
reset [*<slot_ID>*]	Reboot the GAME image on the specified slot. If the slot ID argument is absent, reboot the entire device.	✔	
restart [*<slot_ID>*]	Restart the GAME image on the specified slot. If the slot ID argument is absent, the GAME image restarts on all slots.	✔	

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
rm *<vol>***:***<filename>* (BCC only)	Remove (delete) the file from the specified volume.	✔	
save {**config**|**aliases**} *<vol>***:***<filename>*	Store the current configuration, alias list, or system event.	✔	
securelogin	Turn SecurID access to the device on and off via Telnet.	✔	
set { *<obj_name>* | *<obj_id>* } .{*<attr_name>*| *<attr_id>* } . {*<inst_id>* } {*<value>*} ... (Technician Interface only)	Modifies data objects in the MIB.	✔	
show *<entity> <option>* **show config -all** (BCC only) **show config** (BCC only)	Display statistical information associated with the specified entity and option. Show the total configuration of the device. Show the total configuration of the current object.	✔	✔
source *<vol>***:***<filename>* (BCC configuration mode only)	Read BCC configuration commands from a text file, dynamically changing the active device configuration.	✔	
source {aliases | env | perm } *<vol>*:*<filename>* (Technician Interface only)	Read a list of aliases, environment variables, or dynamic permissions from a file.		
stamp	Display the device image version name and timestamp.	✔	✔
system (Technician Interface only)	Start a new Technician Interface session that allows you to run system manager privileged commands.	✔	✔
telnet [**-d**] [**-e** *<escape_char>*] [*<host_ip>* [*<port>*]] (Technician Interface and BCC top level only; not available in BCC configuration mode)	Communicate with other hosts supporting the Telnet protocol.	✔	✔

(continued)

Table A-1. System Commands *(continued)*

Command	Purpose	Login/Access Privileges	
		Manager	**User**
tftp {**get**|**put**} *<address> <vol>:<file_name>* [*<vol>:<file_name>*] (Technician Interface and bcc> top level only)	Send files to, or retrieve files from, other hosts supporting TFTP.	✔	
type [**-x**] *<vol>:<filename>*	Display the contents of the designated file in ASCII or HEX (**-x**) format.	✔	✔
xmodem {**rb**|**sb}** [**y**|**w**|**p**|**n**] *<filename>* **...**	Transfer files to or from this device over a dial (out-of-band) connection.	✔	

For more detailed information about a specific command:

- Enter **help** <*command*> at any BCC prompt (for example, **help save**).
- Refer to the guide *Using Technician Interface Software.*

Appendix B
TCL Support

The BCC supports the following subset of Tool Command Language (TCL) scripting commands on the router platform:

- append
- break
- case
- catch
- close
- concat
- continue
- eof
- error
- eval
- exit
- expr
- flush
- for
- foreach
- gets
- global
- if
- incr
- interp
- join
- lappend
- lindex
- linsert
- list
- llength
- lrange
- lreplace
- lsearch
- lsort
- open
- proc
- puts
- rename
- return
- set
- source
- split
- subst
- switch
- unset
- uplevel
- upvar
- while

For more information about definitions, syntax, and applications for these TCL commands, refer to the following book:

Ousterhout, J. *Tcl and the Tk Toolkit*. Reading, Mass.: Addison-Wesley, ISBN 0-201-63337-X.

Appendix C
System-Oriented Show Commands

The BCC supports five system-oriented **show** commands, as follows:

- **show config**
- **show console**
- **show hardware**
- **show process**
- **show system**

This appendix describes all BCC system-oriented show commands, except for **show config**. The section "Displaying Configuration Data" in Chapter 2 describes the **show config** command.

show console

The **show console** *<option>* commands display console port configuration and statistics information.

The **show console** command supports the following subcommand options:

config	stats

config

Displays serial port configuration and Technician Interface environment information for the serial ports configured on your system or for a specific port. Entered without an option, this subset of commands displays the total number of ports configured on the system.

The table includes the following information:

Port Number	Port number for the information being displayed. Valid ports are 1, 2, 3, and 4. Not all systems have four physical ports. A configured port that doesn't exist is in the Absent state.
Port State	Port's current state, as follows: • *Absent* - Not physically present • *Disabled* - Unavailable • *Down* - Unavailable • *Init* - Initializing • *Up* - Available
Port Name	Name that the system has given to the port. Users may not specify a name. You can use the name to correlate a port number to a physical port. The name of the port should be printed next to the physical port connection, as follows: • *Port 1* - CONSOLE The names do not specify the port's use. All ports are serial ports used for Technician Interface sessions only. For example, port MODEM1 may be a modem connection or a dummy terminal connection depending on its configuration. While port 4 is called PRINTER, it is exactly like ports 1, 2, and 3. Port 4 doesn't support a printer. It has that name because that's what is printed near the port connector on the link module.

Slot Number	Slot on which the login session for the serial port is running.
Baud Rate	Current baud rate setting for the serial port.
Data Bits	Number of data bits in the serial port's configuration.
Parity	Serial port's current parity setting.
Stop Bits	Number of stop bits in the serial port's configuration.
Modem Control	Configuration of modem control, as follows: • *Disabled* - Port is directly connected to a device, such as a dummy terminal or a terminal server. • *Enabled* - Port is attached to a modem and modem leads are enabled.
Lines/Screen	Number of lines the serial port displays before displaying the more prompt.
More Enable	Setting of the Technician Interface **more** feature: Enabled or Disabled (according to the MIB record). The Technician Interface **more** command affects only the current login session, doesn't change the MIB, and therefore doesn't affect the setting of this field.
Port Prompt	Technician Interface prompt.
Login Retries	Maximum number of login retries; relevant only if modem control is enabled. This value determines the maximum number of failed login attempts a system allows on the serial port. If the maximum occurs, the system hangs up on the line, causing a modem connection to lose carrier detect.
Login Timeout (min.)	Number of minutes allowed between when the system displays the login banner and a user enters a login ID; relevant only if modem control is enabled. If this timeout occurs, the system hangs up on the line.
Password Time Out	Number of minutes allowed for a user to enter a password. If this timeout period occurs, the system hangs up on the line.
Command Time Out	Command line timeout value; relevant only if modem control is enabled. If someone doesn't enter a command in this number of minutes, the system hangs up on the serial port.
User Abort Logout	Switch to execute control (^C) to break out of the user autoscript. When a user autoscript is in effect and this parameter is enabled, you can break out of the script when logged in as User, not Manager. Also, if this parameter is enabled and the script terminates due to an error, the system automatically logs you out.
Initial Search Path	List of file system volumes to be searched when you run a script without a volume specifier or if an autoscript does not contain a volume specifier. The environment variable PATH is set to this string. The string format is as follows: *<vol>*:[;*<vol>*: ...] Example: 2:;3:;4:;5:

Manager's AutoScript	Name of the script to run when the Manager account logs in to the router. If the script name does not contain a volume specifier, the system searches the volumes listed in "Initial Search Path."
User's AutoScript	Name of the script to run when the User account logs in to the router. If the script name does not contain a volume specifier, the system searches the volumes listed in "Initial Search Path."

stats

Displays login information and console port error statistics.

The table includes the following information:

Port Number	Port number for the information displayed.
Port Name	Corresponding port name.
Port State	Port's current state : Absent, Disabled, Down, Init, or Up.
Total Logins	Number of logins (failed and successful) on the console port.
User Login Errors	Number of failed login attempts the User login account has made on the console port.
Manager Login Errors	Number of failed login attempts the Manager login account has made on the console port.
Other Login Errors	Number of failed login attempts made by login accounts other than User and Manager on the console port.
TTY Frame Errors	Number of frame errors on the console port
TTY Overrun Errors	Number of overrun errors on the console port.
TTY Parity Errors	Number of parity errors on the console port.
TTY FIFO Errors	Number of FIFO errors on the console port.

A high number of errors over a short period of time indicates a possible problem with the line.

show hardware

The **show hardware** *<option>* commands display information about a router's hardware.

The **show hardware** command supports the following subcommand options:

backplane	memory
config_file	proms
image	slots

backplane

Displays information about the state of the backplane hardware. The table includes the backplane type, revision, and serial number. The revision and serial numbers are in decimal format.

config_file

Displays the configuration file used to boot the router or reset a slot. The table shows the name and volume that was the source of the configuration as well as the slot that delivered the configuration file to the reset slot. All slots should be running the same configuration file originating from the same volume or backup volume.

image

Displays the router's software image for each slot, including the integration that is the source of the image, the date and time of the image's creation, and the filename that contains the image. If the filename is too long to fit in a horizontal table, the script displays the information in a vertical table, similar to the one shown for the **config_file** command.

memory

Displays memory configuration and capacity information about all slots or a specific slot. The table includes the following information:

Slot	Slot number.
Local Memory	Total memory capacity in megabytes of the processor on the slot.
Global Memory	Current memory configuration in megabytes of the processor on the slot.
Total Memory	Total in megabytes of local and global memory.

proms

Displays PROM information for all slots. The table includes the revision and build date of the Bootstrap PROM and the Diagnostics PROM.

slots

Displays hardware information about all slots in the system. The table includes information about the processor module and link module for each slot, as well as the module type, revision, and serial number. The revision and serial numbers are in decimal format.

For the AN, the table indicates that the AN has an 802.3 repeater (HUB) by indicating that the link module is an ANSEDSH.

For the ASN, the table displays the revision and serial number of the chassis, processor module, and type of the network module type, revision, and serial number.

show process

The **show process** *<option>* commands display information about the use of resources (buffers, cpu, memory, lists, etc.) on the router.

The **show process** command supports the following subcommand options:

buffers [detail|total]	list
cpu [detail|total]	memory

buffers [detail|total]

Displays the number and percentage of buffers used by all processes on the router.

show process buffers [detail|total]

The table includes the following information:

Name	Name of process (“detail only”)
Slot	Slot number.
Used	Number of buffers used
%Used	Percentage of buffers used
Max	Maximum buffers (“total” only)

cpu [detail|total]

Displays the use in 100ths of seconds and percentage of cpu used by all processes on the router. By specifying the options that appear in the following command line, you can customize the output to display cpu usage for a particular process or slot on the router.

show process cpu [**detail| total**]

detail	Displays details about cpu usage by all processes.
total	Displays only cpu statistic totals.

Note: For information on how to obtain a list of available processes, refer to the **show process list** command later in this section.

The table includes the following information:

Name	The name of the
Slot	Slot number.
Used	100ths of seconds used by each process
%Used	Percentage of cpu time used by each process

memory

Displays the number and percentage of memory used by all processes on the router. The output can be tailored to only display memory usage for a particular process or a particular slot by using the options described below.

show process memory [**detail| total**]

The table includes the following information:

detail	Displays details about the amount of memory used by each process running on the router.
total	Displays only memory usage totals.

Name	The name of the process
Slot	Slot number.
Used	Number of memory bytes used by each process
%Used	Percentage of memory used by each process

list

Displays a list of all the processes running on each slot.

show system

The **show system** *<option>* commands display information about the system state, which pertains to the overall system and not to any specific protocol.

The **show system** command supports the following subcommand options:

buffers	memory
drivers	protocols
information	tasks

buffers

Displays the current buffer usage for all active slots in the system. Note that buffers circulate rapidly through the system. A low free percentage doesn't necessarily indicate a buffer shortage; it may be a transient condition.

The table includes the following information:

Slot	Slot number.
Used	Number of buffers used by each process
%Free	Percentage of buffers free on each slot.
Total	Total number of buffers available.
Free	Total number of buffers free.

drivers

Displays link modules and drivers installed on all slots. If the configuration displayed differs from that expected, your configuration file may be incorrect (wrong module type specified; for example) or there may be a problem loading the software.

information

Displays general system information (System name, Contact, node Location, Image data, MIB Version, and total uptime since last cold boot.

memory

Displays the global memory usage for all active slots in the system. Memory usage is not as volatile as buffer usage and a low free percent may indicate that you need more memory.

The table includes the following information:

Slot	Slot number.
Total	Total number of memory (DRAM) bytes available on each slot.
Used	Number of memory bytes used on each slot
Free	Amount of free (unused) memory on each slot.
%Free	Percentage of free (unused) memory on each slot.

protocols

Displays which protocols are installed on all slots in the system. If the configuration displayed differs from that expected, your configuration file may be incorrect (wrong protocol specified, for example) or there may be a problem loading the software.

tasks

Displays the number of tasks scheduled to run on all slots. This number is highly volatile and a large %In Queue does not necessarily indicate a problem.

The table includes the following information:

Slot	Slot number.
Total	Total number of tasks running on each slot.
In_Queue	Number of tasks scheduled to run.
Out_Queue	Number of tasks already run.

Index

A

abbreviating commands and parameters, 1-9

acronyms, ix

ASCII files, 2-27, 3-4

B

backplane, hardware, C-5

Bay Networks Press, x

BCC
- accessing, 1-1, 2-1
- configurable objects, 1-3
- configuration cycle, 4-4
- console types, 2-1
- defined, 1-1
- error messages, 4-19
- Help, 2-12, A-5
- navigating in configuration mode, 2-7
- platform requirements, 1-2
- terminology and concepts, 1-3

board, defined, 1-6

Bootstrap Protocol (BootP)
- PROM, revision date, C-6

box-wide objects, 1-6

buffers, system, C-10

C

class, defined, 1-3

command line
- adding comments to, 3-6
- continuing, 3-4

command operators
- defined, 3-11
- delete, 3-12, 4-18
- disable, 3-12, 4-16
- enable, 3-12, 4-17
- info, 3-12

commands
- ?, 2-12
- back, 2-7
- config, 1-2
- configuration, 3-7
- editing, 2-13, 3-13
- entering, 3-1
- Help, 2-12
- info, 3-12
- list of all system, 3-6, A-1
- lso, 2-23, 2-25
- pwc, 2-6
- show config, 2-23, 2-25, 2-26
- source, 3-3

configuration
- abbreviated syntax, 3-9
- commands, 3-7
- console, C-2
- context-specific, 2-25
- data, 2-23
- default syntax, 3-8
- file, hardware, C-5
- full syntax, 3-7
- hierarchy, 1-4
- method of, 4-2
- new, 4-1
- total, 2-23

connector, 1-8

console subcommands, C-2

context
changing, 2-7
current, 2-6
defined, 1-5, 2-5
displaying, 2-6
context-sensitive prompt, 2-6
continuing a command line, 3-4
conventions, viii
creating a new configuration, 4-1
customer support
programs, x
Technical Solutions Centers, xi

D

delete command (operator), 3-12, 4-18
derived parameters, 1-3
Diagnostics PROM, revision date, C-6
disable command (operator), 3-12, 4-16

E

editing commands, 3-13
enable command (operator), 3-12, 4-17
entering the BCC, 2-1
error messages, 4-19
exiting the BCC, 2-1

G

global objects, defined, 1-6

H

hardware information, C-5
hardware subcommands, C-5
has parameter
example, 2-25
Help
commands, 2-12, A-5
overview, 2-12
history buffer, 3-2

I

image, software, C-5
info command, 3-12
information, show system command, C-11
instance identifier, 1-4, 2-6, 2-9, 3-11
interface, defined, 1-9

L

line, defined, 1-8
list
command operators, A-1
command symbols or conventions, A-5
configurable objects, A-5
configured objects, A-7
editing commands, 3-13
parameter definitions, 2-13
statistical information, A-9
system commands, A-1
location in configuration hierarchy, 2-6
login privileges and procedures, 2-2
lso command, 2-23, 2-25

M

Manager/User privilege levels, 2-2
memory
hardware, C-6
show system command, C-11
messages, error, 4-19
methods, 3-12
modifying a configuration with file-based commands, 4-15

N

navigating
back one or more levels, 2-8
forward one level, 2-9
using configuration commands, 2-7
using the back command, 2-7

O

objects
- defined, 1-3
- deleting, 4-18
- disabling, 4-16
- enabing, 4-17

online Help, BCC, 2-12, A-5

optional attributes, 1-4

P

parameters
- abbreviating, 1-9
- defined, 1-3
- derived values, 1-3
- getting Help, 2-13
- help definitions for, A-5
- required and optional, 3-10
- specifying multiple, 3-11
- specifying values for, 3-10

path, specifying, 2-9, 2-10

port, defined, 1-8

PROM information, hardware, C-6

prompt, context-sensitive, 2-6

protocol, defined, 1-9

protocols, show system command, C-11

publications, ordering, x

pwc command, 2-6

R

reading commands from a file, 3-3

recalling commands, 3-2

required attributes, defined, 1-3, 3-8

S

saving configuration commands, 2-27

serial port
- configuration, C-2
- error statistics, C-4

show commands
- console, C-2
- hardware, C-5
- system, C-10

show config
- command, 2-23
- example, 2-25

slot, defined, 1-6

slots
- hardware, C-6

source command
- defined, 3-3, 4-15
- syntax, 3-4

specifying a path, 2-9, 2-10

statistics
- console, C-4

syntax
- getting Help on, 2-12
- types, 3-1

system information, C-11

SYSTEM subcommands, C-10

T

tasks, show system command, C-12

TCL support, B-1

Technical Solutions Centers, xi

Technician Interface, 1-1, 1-2

Technician Interface environment, C-2

U

User privileges, 2-2